NAVAL POSTGRADUATE SCHOOL
Monterey, California

DISSERTATION

ELECTROMAGNETIC SCATTERING OF AN ANISOTROPICALLY COATED TUBULAR CYLINDER

by

Chen-Kuo Yu

March, 1997

Dissertation Supervisor: Hung-Mou Lee

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE March 1997.	3. REPORT TYPE AND DATES COVERED Ph.D. Dissertation

4. TITLE AND SUBTITLE ELECTROMAGNETIC SCATTERING OF AN ANISOTROPICALLY COATED TUBULAR CYLINDER	5. FUNDING NUMBERS
6. AUTHOR(S) Chen-Kuo Yu	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)* The sum-difference surface current formulation is introduced to treat electromagnetic boundary value problems when anisotropic impedances are specified on both sides of a surface. It can also be applied to impedance coated bodies. This formulation preserves the duality nature of Maxwell equations and carries it over into the algebraic form of the integrodifferential operators in the equations for surface currents. Since a 90° rotation is equivalent to undergoing a duality transform for an incident plane wave, this particular symmetry in the algebraic form of the operators leads to sufficient conditions under which the on-axis backscattering of an anisotropic impedance coated scatterer having a 90° rotational symmetry is eliminated. The sum-difference formulation is utilized for solving the problem of electromagnetic scattering from an anisotropically impedance coated tubular cylinder of finite length. The solution has been coded in FORTRAN and tested. Some interesting results are presented and discussed.

14. SUBJECT TERMS zero backscattering cross section, impedance boundary condition, anisotropic surface impedance	15. NUMBER OF PAGES 127
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18 298-102

ELECTROMAGNETIC SCATTERING
OF AN ANISOTROPICALLY COATED TUBULAR CYLINDER

Chen-Kuo Yu
Captain, Republic of China Navy
B.S., Chinese Naval Academy, 1976
M.S.S.E., Naval Postgraduate School, 1990

Submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY IN (Electrical Engineering)

from the

NAVAL POSTGRADUATE SCHOOL
March 1997

Abstract

The sum-difference surface current formulation is introduced to treat electromagnetic boundary value problems when anisotropic impedances are specified on both sides of a surface. It can also be applied to impedance coated bodies. This formulation preserves the duality nature of Maxwell equations and carries it over into the algebraic form of the integrodifferential operators in the equations for surface currents. Since a 90° rotation is equivalent to undergoing a duality transform for an incident plane wave, this particular symmetry in the algebraic form of the operators leads to sufficient conditions under which the on-axis backscattering of an anisotropic impedance coated scatterer having a 90° rotational symmetry is eliminated. The sum-difference formulation is utilized for solving the problem of electromagnetic scattering from an anisotropically impedance coated tubular cylinder of finite length. The solution has been coded in FORTRAN and tested. Some interesting results are presented and discussed.

TABLE OF CONTENTS

I. INTRODUCTION

Sometime ago the question was raised: "For electromagnetic boundary value problems with specified surface impedances, how can one go from a non-perfectly conducting surface on which both the electric and the magnetic equivalent surface currents are to be found, to a perfectly conducting surface on which the number of unknowns is halved [1]?" The answer to this question turns out to be one of algebra. It is well known that the impedance specified on the surface of a body separates its interior completely from its exterior. Therefore an impedance coated body can always be considered as a hollow volume enclosed by an infinitesimally thin shell with surface impedances specified both on the inside and the outside of the shell. The inside and the outside of the body can be considered as constituted of the same medium and the impressed electromagnetic excitation can be treated as continuous across the shell. On the outside surface, there are the equivalent total electric current $\vec{K}^+$ and total magnetic current $\vec{L}^+$; on the inside surface, there are $\vec{K}^-$ and $\vec{L}^-$. For an exterior problem, only $\vec{K}^+$ and $\vec{L}^+$ need to be found; for an interior problem, only $\vec{K}^-$ and $\vec{L}^-$ are necessary. A single formulation for solving both types of problems would appear to require finding all inside and outside currents therefore doubling the amount of work, but it turns out not to be the case because some of the currents are linear combinations of others. Furthermore, this formulation holds the key to answering the question posed above.

Since the shell is infinitesimally thin, from Maxwell equations the radiation to the outside and to the inside of the shell can both be given in terms of integrodifferential operators on the sum currents $\vec{K} = \vec{K}^+ + \vec{K}^-$ and $\vec{L} = \vec{L}^+ + \vec{L}^-$. Note that the outside currents will not contribute to the radiation in the interior while the inside currents will not contribute to the radiation to the exterior of the shell. For simplicity in the description, we consider the exterior problem of electromagnetic scattering. By definition, the radiation is the difference between the total field and the incident field. On the surfaces of the shell, this definition links the incident $\vec{E}$ and $\vec{H}$ fields and the difference currents $\vec{K}^+ - \vec{K}^-$ and $\vec{L}^+ - \vec{L}^-$ to the sum currents. It is therefore natural to treat the difference currents and the sum currents as the four unknowns to be solved instead of the inside and the outside currents.

The surface impedance on the outside surface of the shell normalized to the medium is denoted by Z^+ and that on the inside surface is Z^-. They can be tensors if the impedances are anisotropic and may vary from point to point. By forming the sum impedance $Z = (Z^+ + Z^-)/2$ and the difference impedance $\Delta = (Z^+ - Z^-)/2$, the impedance boundary conditions provide a set of relations between the difference and the sum currents. It turns out that the rank of Z determines how the unknown surface currents are solved. If Z is invertible, then the difference currents are linear combinations of the sum currents so that only the integrodifferential equations of the sum currents have to be solved. There are only two unknowns to be solved for both the exterior and the interior problems under this sum-difference current formulation. If Z is rank 0, then Z = 0; the impedance boundary condition requires that $\vec{L}$ be proportional to a $90°$ rotation of $\Delta\vec{K}$. The difference electric current is obtained from $\vec{K}$ after the integrodifferential equation on $\vec{K}$ is solved. This situation includes the case where $Z^+ = Z^- = 0$ when the surface is perfectly conducting, thus the result answers the question about the transition of the equations from a problem of two variables to one which has only a single variable.

Instead of dealing with an impedance coated body, this thesis presents the sum-difference currents formulation of electromagnetic boundary value problems for the scattering of an infinitesimally thin surface for which both the inside and the outside currents are true unknowns to be found. Extension of this formulation to impedance coated bodies is then discussed.

This formulation preserves the duality nature of Maxwell equations and carries it over into an explicit specific algebraic form of the integrodifferential operators in the equations for the sum currents. Since, for a plane wave, a $90°$ rotation is equivalent to undergoing a duality transform, this explicit symmetry in the algebraic form of the operators enables us to deduce sufficient conditions for eliminating the on-axis backscattering of an anisotropic impedance coated scatterer having a $90°$ rotational symmetry.

The sum-difference currents formulation is utilized for solving the problem of electromagnetic scattering from an anisotropically impedance coated tubular cylinder of finite length. The solution has been coded in FORTRAN and tested. Some interesting results are presented and discussed.

In this thesis, the time dependence $e^{-i\omega t}$ is used. $\vec{E}$ represents the electric field intensity divided by the intrinsic impedance of the medium, $\sqrt{\mu/\epsilon}$; therefore it has the same unit as $\vec{H}$ in amperes per meter. So are the electric and magnetic equivalent surface currents $\vec{K}$ and $\vec{L}$.

II. THE SUM-DIFFERENCE SURFACE CURRENT FORMULATION OF ELECTROMAGNETIC BOUNDARY-VALUE PROBLEMS

A. STRATTON-CHU FIELD FORMULATION AND RADIATION

On an orientable, piecewise regular surface, whether open or closed, having a surface electric current density $\vec{K}$ and a surface magnetic current density $\vec{L}$, we define the Stratton-Chu E-field formula as:

$$\begin{aligned}\vec{E}_{s-c}(\vec{r}, S, \vec{K}, \vec{L}) = & \frac{ik^2}{4\pi}\int_S \vec{K}(\vec{r}_o)G(\vec{r}-\vec{r}_o)da_o - \frac{i}{4\pi}\nabla\int_S \vec{K}(\vec{r}_o)\bullet\nabla_o G(\vec{r}-\vec{r}_o)da_o \\ & - \frac{k}{4\pi}\nabla\times\int_S \vec{L}(\vec{r}_o)G(\vec{r}-\vec{r}_o)da_o\end{aligned} \tag{2-1}$$

where $\vec{r}$ is a point which is not on S, $k = \omega\sqrt{\mu_o\varepsilon_o}$ and $G(\vec{r}-\vec{r}_o) = \dfrac{e^{ik|\vec{r}-\vec{r}_o|}}{k|\vec{r}-\vec{r}_o|}$, then the Stratton-Chu H-field formula can be defined as:

$$\begin{aligned}\vec{H}_{s-c}(\vec{r}, S, \vec{K}, \vec{L}) &= \vec{E}_{s-c}(\vec{r}, S, \vec{L}, -\vec{K}) \\ &= \frac{k}{4\pi}\nabla\times\int_S \vec{K}(\vec{r}_o)G(\vec{r}-\vec{r}_o)da_o + \frac{ik^2}{4\pi}\int_S \vec{L}(\vec{r}_o)G(\vec{r}-\vec{r}_o)da_o \\ &\quad - \frac{i}{4\pi}\nabla\int_S \vec{L}(\vec{r}_o)\bullet\nabla_o G(\vec{r}-\vec{r}_o)da_o\end{aligned} \tag{2-2}$$

Note that if S is a closed surface and $\vec{K}$ and $\vec{L}$ are the actual total equivalent surface currents on S, then $\vec{E}_{s-c}$ and $\vec{H}_{s-c}$ are respectively the $\vec{E}$ and $\vec{H}$ fields at $\vec{r}$ due to all sources inside S if $\vec{r}$ is located outside S and vice versa. This is a direct consequence of Maxwell's equations [2] and under this circumstance, the Stratton-Chu formulae are equivalent to Maxwell's equations. On the other hand, unlike the Poynting theorem, the Stratton-Chu field formulae over an open surface S are but integrodifferential operators on the tangential vector fields $\vec{K}$ and $\vec{L}$ over S without any special physical meaning attached.

To introduce equivalent surface currents on S, the direction of the unit normal vector $\hat{n}$ on the surface has to be chosen. Adopting the terminology for a closed surface, we can assign one side of any orientable surface S as the "outside surface" S^+, albeit somewhat arbitrarily if S is not closed. The outward normal $\hat{n}^+$ is the unit normal pointing out of this side of S and for simplicity, we call $\hat{n}^+$ the normal of S and denote it by $\hat{n}$. The other side of the surface S is the "inside surface" S^-. At any point $\vec{r}$ on S, the inward normal $\hat{n}^-$ is $-\hat{n}$. As a convention, the fields and surface currents on S^+ and S^- always carry the corresponding superscripts (Figure 2-1).

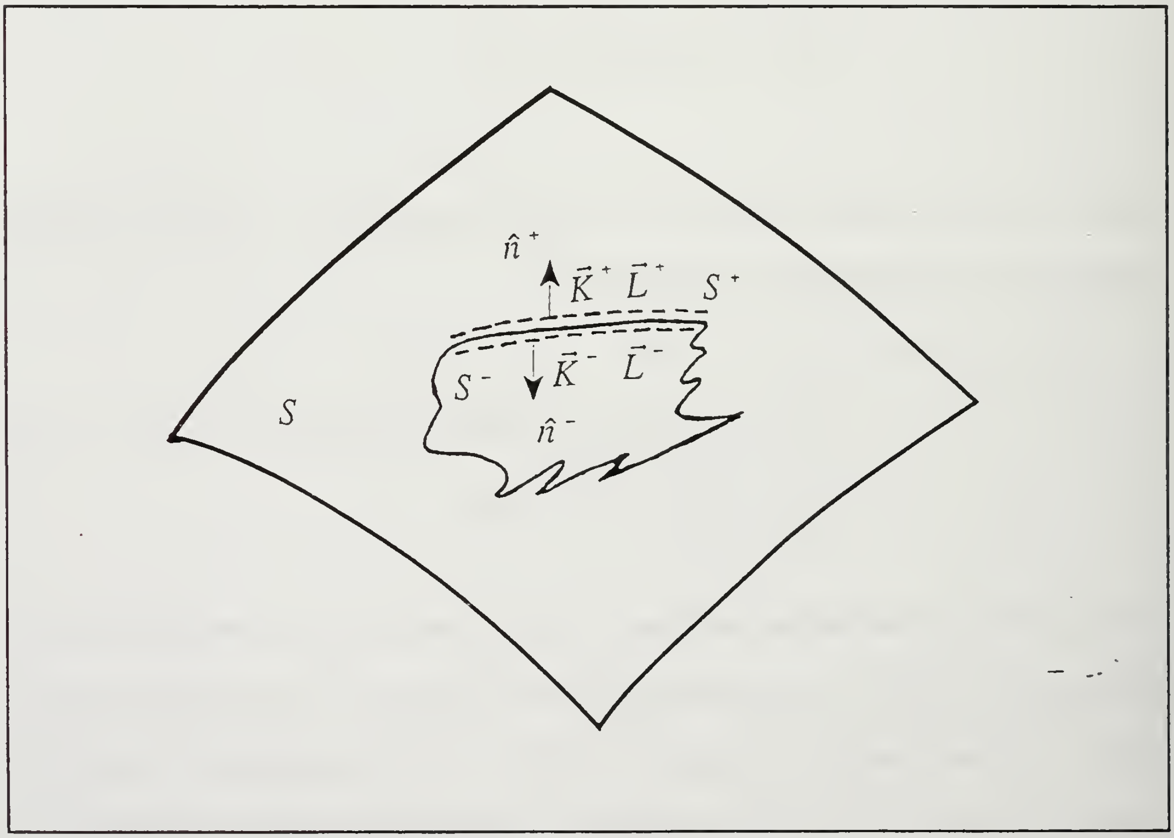

Figure 2-1. Outside and Inside Surfaces, Normals and the Equivalent Currents.

Each of the total surface currents $\vec{K}^\pm$ and $\vec{L}^\pm$ on $S^\pm$ consists of two parts: the incident current (with the additional superscript "*inc*") and the scattering current (with the additional superscript "*sc*") corresponding to the incident field and the scattered field on the particular side of the surface S:

$$\begin{aligned}\vec{K}^{\pm} &= \hat{n}^{\pm} \times \vec{H}^{\pm} = \hat{n}^{\pm} \times \vec{H}^{inc} + \hat{n}^{\pm} \times \vec{H}^{\pm,sc} \\ &= \vec{K}^{\pm,inc} + \vec{K}^{\pm,sc}\end{aligned} \tag{2-3}$$

$$\begin{aligned}\vec{L}^{\pm} &= \vec{E}^{\pm} \times \hat{n}^{\pm} = \vec{E}^{inc} \times \hat{n}^{\pm} + \vec{E}^{\pm,sc} \times \hat{n}^{\pm} \\ &= \vec{L}^{\pm,inc} + \vec{L}^{\pm,sc}\end{aligned} \tag{2-4}$$

Note that S is infinitesimally thin, hence $\vec{H}^{+,inc} = \vec{H}^{-,inc} = \vec{H}^{inc}$ and $\vec{E}^{+,inc} = \vec{E}^{-,inc} = \vec{E}^{inc}$ on S so that $\vec{K}^{+,inc} = -\vec{K}^{-,inc}$ and $\vec{L}^{+,inc} = -\vec{L}^{-,inc}$. Therefore the sum currents $\vec{K}$ and $\vec{L}$ on S defined below are also the corresponding sums of the scattering currents only:

$$\begin{aligned}\vec{K} &= \vec{K}^{+} + \vec{K}^{-} = \vec{K}^{+,sc} + \vec{K}^{-,sc} \\ \vec{L} &= \vec{L}^{+} + \vec{L}^{-} = \vec{L}^{+,sc} + \vec{L}^{-,sc}\end{aligned} \tag{2-5}$$

Since the Stratton-Chu field formulae are linear operators on the surface currents, the radiated fields from surface currents on S are determined by the sum currents only:

$$\begin{aligned}\vec{E}^{sc}(\vec{r}) &= \vec{E}_{s\text{-}c}(\vec{r},S^{+},\vec{K}^{+},\vec{L}^{+}) + \vec{E}_{s\text{-}c}(\vec{r},S^{-},\vec{K}^{-},\vec{L}^{-}) = \vec{E}_{s\text{-}c}(\vec{r},S,\vec{K},\vec{L}) \\ \vec{H}^{sc}(\vec{r}) &= \vec{H}_{s\text{-}c}(\vec{r},S,\vec{K},\vec{L}) = \vec{E}_{s\text{-}c}(\vec{r},S,\vec{L},-\vec{K})\end{aligned} \tag{2-6}$$

B. CONDITION ON THE CURRENTS IMPOSED BY MAXWELL'S EQUATIONS

As $\vec{r}$ approaches $\vec{r}^{\pm}$ on $S^{\pm}$, the tangential components (denoted by the subscript "*tan*") of Eq. (2-6) provide four equations relating the incident fields and the total currents on both sides of S through the fact that the incident field is the difference between the total and the scattered field:

$$\vec{E}_{\tan}^{inc} = \vec{E}_{\tan}^{+} - \vec{E}_{s-c,\tan}(\vec{r}^{+}, S, \vec{K}, \vec{L}) \tag{2-7}$$

$$\vec{E}_{\tan}^{inc} = \vec{E}_{\tan}^{-} - \vec{E}_{s-c,\tan}(\vec{r}^{-}, S, \vec{K}, \vec{L}) \tag{2-8}$$

$$\vec{H}_{\tan}^{inc} = \vec{H}_{\tan}^{+} - \vec{H}_{s-c,\tan}(\vec{r}^{+}, S, \vec{K}, \vec{L}) \tag{2-9}$$

$$\vec{H}_{\tan}^{inc} = \vec{H}_{\tan}^{-} - \vec{H}_{s-c,\tan}(\vec{r}^{-}, S, \vec{K}, \vec{L}) \tag{2-10}$$

These four equations are not independent of each other. Because

$$\begin{aligned} \hat{n}^{\pm} \times (\vec{E}(\vec{r}^{\pm}) \times \hat{n}^{\pm}) &= \vec{E}_{\tan}(\vec{r}^{\pm}) = \hat{n}^{\pm} \times \vec{L}^{\pm} \\ \hat{n}^{\pm} \times (\vec{H}(\vec{r}^{\pm}) \times \hat{n}^{\pm}) &= \vec{H}_{\tan}(\vec{r}^{\pm}) = -\hat{n}^{\pm} \times \vec{K}^{\pm} \end{aligned} \tag{2-11}$$

the difference between Eqs. (2-7) and (2-8) trivially confirms the definition of the sum equivalent magnetic current while the difference between Eqs. (2-9) and (2-10) confirms the definition of the sum equivalent electric current as both can be deduced directly from Maxwell's equations. We choose to use the sum of Eqs. (2-7) and (2-8) and that of Eqs. (2-9) and (2-10) as the two independent linear combinations out of Eqs. (2-7) through (2-10) to link the incident fields to the total surface currents on $S^{\pm}$ as dictated by Maxwell's equations:

$$
\begin{aligned}
2\vec{E}^{\,inc}_{\tan} &= \hat{n} \times (\vec{L}^{+} - \vec{L}^{-}) - \left\{\vec{E}_{s-c}(\vec{r}^{\,+},S,\vec{K},\vec{L}) + \vec{E}_{s-c}(\vec{r}^{\,-},S,\vec{K},\vec{L})\right\} \\
&= \hat{n} \times (\vec{L}^{+} - \vec{L}^{-}) + M\vec{K} - N\vec{L} \\
2\vec{H}^{\,inc}_{\tan} &= -\,\hat{n} \times (\vec{K}^{+} - \vec{K}^{-}) - \left\{\vec{H}_{s-c}(\vec{r}^{\,+},S,\vec{K},\vec{L}) + \vec{H}_{s-c}(\vec{r}^{\,-},S,\vec{K},\vec{L})\right\} \\
&= -\,\hat{n} \times (\vec{K}^{+} - \vec{K}^{-}) + N\vec{K} + M\vec{L}
\end{aligned}
\tag{2-12}
$$

where M and N are linear integrodifferential operators on the tangential vector fields $\vec{K}$ and $\vec{L}$ over S.

Under any orthonormal coordinates (u, v) over S having $\hat{u}$, $\hat{v}$ as the unit basis vectors and with $\hat{n} = \hat{u} \times \hat{v}$, a tangential vector field $\vec{A}$ over S can be written in matrix form as: $\vec{A} = \begin{bmatrix} A_u \\ A_v \end{bmatrix}$. Then $\hat{n} \times \vec{A} = \begin{bmatrix} -A_v \\ A_u \end{bmatrix} = -i\sigma_2 \vec{A}$ where $\sigma_2 = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$ is one of the Pauli spin matrices. Note that $\sigma_2^{\,2} = 1$. Using these matrix notations, we can rewrite Eq. (2-12) in the following form:

$$
\begin{bmatrix} 0 & i\sigma_2 \\ -i\sigma_2 & 0 \end{bmatrix} \begin{bmatrix} \vec{K}^{+} - \vec{K}^{-} \\ \vec{L}^{+} - \vec{L}^{-} \end{bmatrix} = \begin{bmatrix} M & -N \\ N & M \end{bmatrix} \begin{bmatrix} \vec{K} \\ \vec{L} \end{bmatrix} - 2 \begin{bmatrix} \vec{E}^{\,inc}_{\tan} \\ \vec{H}^{\,inc}_{\tan} \end{bmatrix}
\tag{2-13}
$$

C. IMPEDANCE BOUNDARY CONDITION

Maxwell's equations alone cannot determine the electromagnetic fields completely. If S is an open surface, appropriate boundary condition which the fields satisfy on S must be specified. It is usually given in terms of the impedance boundary condition, a linear relation among the tangential components of the total $\vec{E}$ and the total $\vec{H}$ fields on S. If S is a closed surface, there are two possibilities: One is to specify the electric and magnetic properties of the volume within S and require the fields to satisfy regularity conditions within S and be linked to the fields outside through boundary conditions across S; another is to specify the impedance boundary condition on S^+ for an exterior problem or on S^- for an interior problem. Note that an impedance boundary condition over a closed surface S completely separates the exterior from the interior of S. Therefore, the surface impedance on S^- can be arbitrary for an exterior problem while that on S^+ can be arbitrary for an interior problem. In this thesis, normalized surface impedances $Z^\pm$ are assumed to be specified on $S^\pm$ whether S is an open or a closed surface. Note that an open surface S can be considered as bounded within the closed surface formed by joining S^+ and S^-.

The impedance boundary conditions on $S^\pm$ are defined by:

$$\hat{n}^\pm \times (\vec{E}^\pm \times \hat{n}^\pm) = Z^\pm (\hat{n}^\pm \times \vec{H}^\pm) \tag{2-14}$$

or equivalently, in terms of the total surface currents:

$$\hat{n}^\pm \times \vec{L}^\pm = Z^\pm \vec{K}^\pm \tag{2-15}$$

With the matrix notations for tangential vector fields over S in the orthonormal coordinate system (u, v) introduced before, we can consider $Z^\pm$ as 2×2 matrices and rewrite Eq. (2-15) in the form:

$$\mp i\sigma_2 \vec{L}^\pm = Z^\pm \vec{K}^\pm = \frac{1}{2} Z^\pm [\vec{K} \pm (\vec{K}^+ - \vec{K}^-)] \tag{2-16}$$

which can readily be recast into a relation among sum and difference currents:

$$\begin{bmatrix} -\Delta & -i\sigma_2 \\ Z & 0 \end{bmatrix} \begin{bmatrix} \vec{K}^+ - \vec{K}^- \\ \vec{L}^+ - \vec{L}^- \end{bmatrix} = \begin{bmatrix} Z & 0 \\ -\Delta & -i\sigma_2 \end{bmatrix} \begin{bmatrix} \vec{K} \\ \vec{L} \end{bmatrix} \tag{2-17}$$

with

$$Z = \frac{1}{2}(Z^{+} + Z^{-}) \tag{2-18}$$

and

$$\Delta = \frac{1}{2}(Z^{+} - Z^{-}) \tag{2-19}$$

Eqs. (2-13) and (2-17) are a set of four two-dimensional vector equations to be solved for the sum and difference equivalent electric and magnetic surface current densities on S.

D. ALGEBRA OF THE SUM-DIFFERENCE CURRENT EQUATIONS

The existence and uniqueness of a solution to either the exterior or the interior problem specified in terms of the impedance boundary condition have been well established [3]. Here we want to investigate how such a solution can be obtained from Eqs. (2-13) and (2-17). Clearly Eq. (2-17) defines uniquely the algebraic relationship between the difference and the sum currents if Z is invertible. For example, the difference currents can be expressed in terms of the sum currents:

$$\begin{bmatrix} \vec{K}^{+} - \vec{K}^{-} \\ \vec{L}^{+} - \vec{L}^{-} \end{bmatrix} = - \begin{bmatrix} 0 & i\sigma_2 \\ -i\sigma_2 & 0 \end{bmatrix} R \begin{bmatrix} \vec{K} \\ \vec{L} \end{bmatrix} \tag{2-20}$$

where

$$R = \begin{bmatrix} Z - \Delta Z^{-1} \Delta & -i \Delta Z^{-1} \sigma_2 \\ -i\sigma_2 Z^{-1} \Delta & \sigma_2 Z^{-1} \sigma_2 \end{bmatrix} \tag{2-21}$$

An equation for the sum currents is obtained by substituting Eq. (2-20) into Eq. (2-13):

$$\begin{bmatrix} M & -N \\ N & M \end{bmatrix} \begin{bmatrix} \vec{K} \\ \vec{L} \end{bmatrix} + R \begin{bmatrix} \vec{K} \\ \vec{L} \end{bmatrix} = 2 \begin{bmatrix} \vec{E}_{\tan}^{inc} \\ \vec{H}_{\tan}^{inc} \end{bmatrix} \tag{2-22}$$

which can be solved for $\vec{K}$ and $\vec{L}$. Eq. (2-20) in turn enables us to compute the difference currents algebraically then split the sum and the difference currents into total currents on $S^{\pm}$.

If Z is not invertible, then the situation is more complicated. Z can either be of rank 0 when $Z = 0$ or rank 1 when $\det Z = 0$ but $Z \neq 0$. Eqs. (2-13) and (2-17) can be combined into an equation for the sum currents $\vec{K}$ and $\vec{L}$:

$$\begin{bmatrix} 1 & i\Delta\sigma_2 \\ 0 & -iZ\sigma_2 \end{bmatrix} \begin{bmatrix} M & -N \\ N & M \end{bmatrix} \begin{bmatrix} \vec{K} \\ \vec{L} \end{bmatrix} + \begin{bmatrix} Z & 0 \\ -\Delta & -i\sigma_2 \end{bmatrix} \begin{bmatrix} \vec{K} \\ \vec{L} \end{bmatrix} = 2 \begin{bmatrix} 1 & i\Delta\sigma_2 \\ 0 & -iZ\sigma_2 \end{bmatrix} \begin{bmatrix} \vec{E}_{\tan}^{inc} \\ \vec{H}_{\tan}^{inc} \end{bmatrix} \tag{2-23}$$

When $Z = 0$, Eq. (2-17) gives the null relations $\vec{L}^{+} - \vec{L}^{-} = i\sigma_2\Delta(\vec{K}^{+} - \vec{K}^{-})$ and $\vec{L} = i\sigma_2\Delta\vec{K}$ hence $\vec{L}^{\pm} = i\sigma_2\Delta\vec{K}^{\pm}$. Eq. (2-23) becomes one for $\vec{K}$ only:

$$\begin{bmatrix} 1 & i\Delta\sigma_2 \end{bmatrix} \begin{bmatrix} M & -N \\ N & M \end{bmatrix} \begin{bmatrix} 1 \\ i\sigma_2\Delta \end{bmatrix} \vec{K} = 2 \begin{bmatrix} 1 & i\Delta\sigma_2 \end{bmatrix} \begin{bmatrix} \vec{E}_{\tan}^{inc} \\ \vec{H}_{\tan}^{inc} \end{bmatrix} \tag{2-24}$$

Eq. (2-13) has to be used to find the difference electric current:

$$\vec{K}^{+} - \vec{K}^{-} = i\sigma_2\left[N + iM\sigma_2\Delta\right]\vec{K} - 2i\sigma_2\vec{H}_{\tan}^{inc} \tag{2-25}$$

Therefore,

$$\vec{K}^{\pm} = \frac{1}{2}\left\{1 \pm i\sigma_2\left[N + iM\sigma_2\Delta\right]\right\}\vec{K} \mp i\sigma_2\vec{H}_{\tan}^{inc} \tag{2-26}$$

Since the last term in Eq. (2-26) is $\vec{K}^{\pm,inc}$,

$$\vec{K}^{\pm,sc} = \frac{1}{2}\left\{1 \pm i\sigma_2\left[N + iM\sigma_2\Delta\right]\right\}\vec{K} \tag{2-27}$$

$\vec{L}^{+}$ and $\vec{L}^{-}$ can be obtained algebraically by multiplying $i\sigma_2\Delta$ to $\vec{K}^{+}$ and $\vec{K}^{-}$ respectively. On the other hand, by Eq. (2-13),

$$\vec{L}^{\pm,sc} = \frac{1}{2}i\sigma_2\left\{\Delta \mp \left[M - iN\sigma_2\Delta\right]\right\}\vec{K} \tag{2-28}$$

Note that the $Z = 0$ case includes the special situation $Z^{+} = Z^{-} = Z = \Delta = 0$ when both sides of S are perfectly conducting. Under this circumstance $\vec{L} = \vec{L}^{\pm} = 0$ and the operator N is never involved.

When Z is rank 1, $Z \neq 0$ but $\det Z = 0$. The right hand side of Eq. (2-17) provides one linear relation between the components of $\vec{L} - i\sigma_2\Delta\vec{K}$ which can be used to reduce the four components of $\vec{K}$ and $\vec{L}$ as the unknown quantities in Eq. (2-23) to three so that the remaining three components can be solved. The left hand side of Eq. (2-17) assures that the same linear relationship between the components of $\vec{L} - i\sigma_2\Delta\vec{K}$ exists between corresponding components of the difference currents. Eq. (2-13) again has to be invoked to compute three other linearly independent combinations of the components of the difference currents from the sum currents.

E. CONSIDERATIONS FOR A CLOSED SURFACE

When S is a closed surface, the choice of Z^- can be arbitrary for an exterior problem such as scattering while the choice of Z^+ is arbitrary for an interior problem. It is desirable to choose $Z^- = -Z^+$ so that $Z = 0$ and $\Delta = Z^+ = -Z^-$. Then we have $\vec{L} = i\sigma_2 \Delta \vec{K}$ and only $\vec{K}$ has to be computed. With such a choice, for an exterior problem,

$$\vec{K}^{+,sc} = \frac{1}{2}(1 - \sigma_2 M \sigma_2 Z^+ + i\sigma_2 N)\vec{K} \tag{2-29}$$

$$\vec{L}^{+,sc} = \frac{i\sigma_2}{2}(Z^+ + iN\sigma_2 Z^+ - M)\vec{K} \tag{2-30}$$

and for an interior problem:

$$\vec{K}^{-,sc} = \frac{1}{2}(1 - \sigma_2 M \sigma_2 Z^- - i\sigma_2 N)\vec{K} \tag{2-31}$$

$$\vec{L}^{-,sc} = \frac{i\sigma_2}{2}(M - Z^- + iN\sigma_2 Z^-)\vec{K} \tag{2-32}$$

III. A THEOREM OF ANISOTROPIC ABSORBERS

A. AXIAL RADIATION FROM A SURFACE OF 90° ROTATIONAL SYMMETRY

The xy-plane cross section of a surface S having a $90°$ rotational symmetry around the z-axis is shown in Figure 3-1. Because of this symmetry, S can be decomposed into four non-overlapping congruent pieces S_1, S_2, S_3, S_4 so that a $90°$ rotation will bring S_i into S_{i+1}.

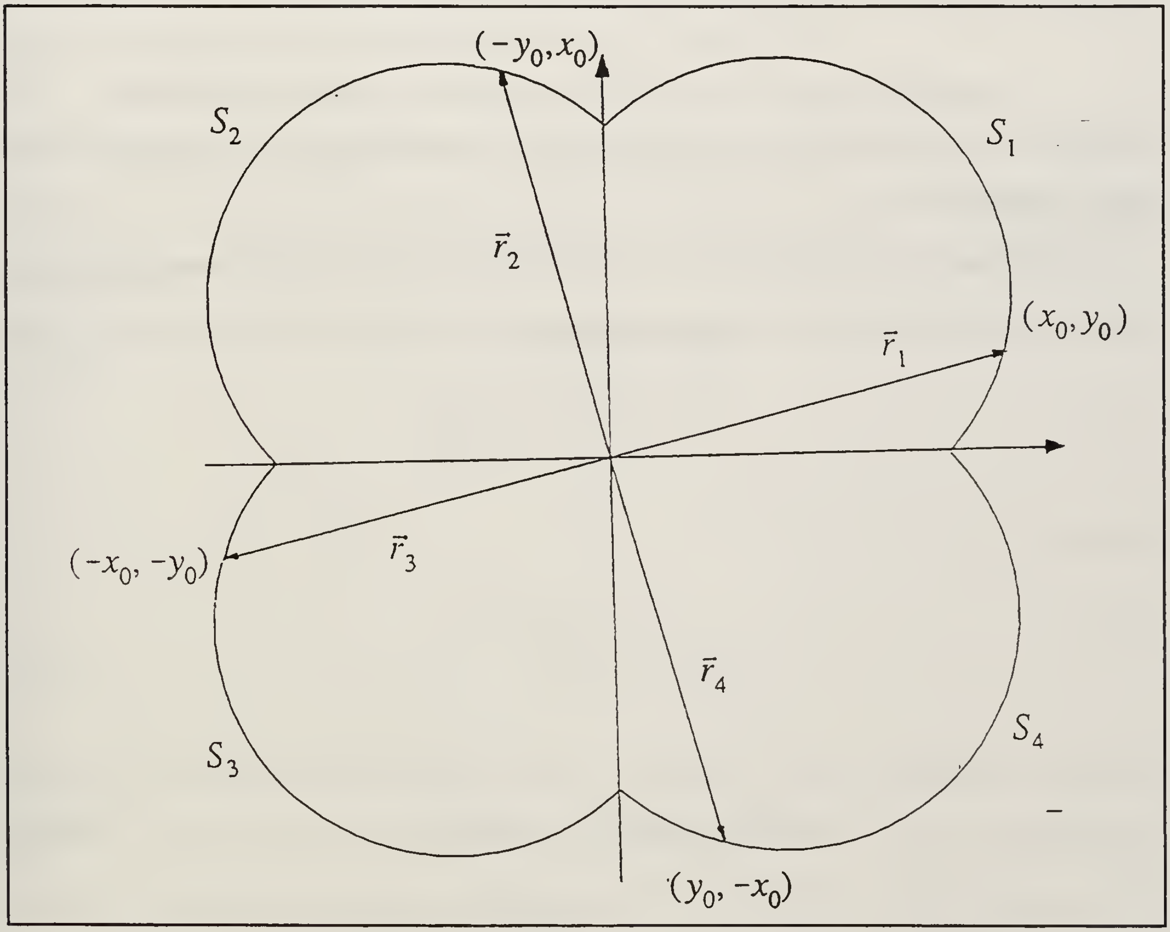

Figure 3-1. Cross Section of a Surface of 90-Degree Rotational Symmetry.

(These subscripts are considered as equal under modulo 4.) Therefore each piece S_i of S can be parametrized in the same orthonormal coordinates (u, v), with $\hat{u} = \frac{\partial \vec{r}_i}{\partial u}$, $\hat{v} = \frac{\partial \vec{r}_i}{\partial v}$ the orthonormal basis vectors on S_i, as follows:

$$\begin{aligned}
\vec{r}_1 &= (x_o(u,v), y_o(u,v), z_o(u,v)) \\
\vec{r}_2 &= (-y_o(u,v), x_o(u,v), z_o(u,v)) \\
\vec{r}_3 &= (-x_o(u,v), -y_o(u,v), z_o(u,v)) \\
\vec{r}_4 &= (y_o(u,v), -x_o(u,v), z_o(u,v))
\end{aligned} \tag{3-1}$$

where $\vec{r}_i \in S_i$. As an example of a possible choice, u = constant and v = constant can be the lines of curvature of S_I.

In terms of the coordinates (u, v), the sum surface current distributions on S_i are:

$$\begin{aligned}
\vec{K}(\vec{r}_i) &= \vec{K}_i(u,v) \\
\vec{L}(\vec{r}_i) &= \vec{L}_i(u,v)
\end{aligned} \tag{3-2}$$

Since for $r \gg r_o$,

$$G(\vec{r}) \approx \frac{e^{ik|\vec{r}-\vec{r}_o|}}{k r} \tag{3-3}$$

$$\nabla G(\vec{r}) \approx ik\hat{r}G \approx -\nabla_o G(\vec{r}) \tag{3-4}$$

the radiation from such current distributions at a distance $r \gg \max |\vec{r}_1|$ along the positive z-axis is, from Eq. (2-6):

$$
\begin{aligned}
\vec{E}^{sc}(\vec{r}) &= \vec{E}_{s-c}(\vec{r},S,\vec{K},\vec{L}) \\
&= \frac{ik}{4\pi r}\int_S \left\{\hat{x}[K_x(\vec{r}_o)+L_y(\vec{r}_o)] + \hat{y}[K_y(\vec{r}_o)-L_x(\vec{r}_o)]\right\} e^{ik\sqrt{(z-z_o)^2+x_o^2+y_o^2}}\, da_o \\
&= \frac{ik}{4\pi r}\int\int_{S_1}\left\{ \hat{x}\sum_{i=1}^{4}[K_{ix}(u,v)+L_{iy}(u,v)] \right. \\
&\qquad \left. + \hat{y}\sum_{i=1}^{4}[K_{iy}(u,v)-L_{ix}(u,v)]\right\} e^{ik\sqrt{(z-z_o)^2+x_o^2+y_o^2}}\, du\, dv
\end{aligned}
\tag{3-5}
$$

Note that this approximation is independent of the wavelength; it is applicable in regions closer to S than the usual Fresnel zone.

B. CONDITION FOR VANISHING ON-AXIS BACKSCATTERING

Consider two situations when a linearly polarized plane wave of unit strength is incident on S along the z-axis from the positive direction: Situation 1, identified with the superscript (1) has the wave polarized in the x-direction while Situation 2, identified with the superscript (2), has the wave polarized in the y-direction. The incident fields are respectively:

$$
\begin{aligned}
\vec{E}^{inc,(1)} &= \hat{x}\, e^{-ikz} \\
\vec{H}^{inc,(1)} &= -\,\hat{y}\, e^{-ikz}
\end{aligned}
\tag{3-6}
$$

$$
\begin{aligned}
\vec{E}^{inc,(2)} &= \hat{y}\, e^{-ikz} \\
\vec{H}^{inc,(2)} &= \hat{x}\, e^{-ikz}
\end{aligned}
\tag{3-7}
$$

Note that, as seen from the positive z-axis, the incident wave in Situation 2 is that of Situation 1 rotated by 90^o counterclockwise. Furthermore, Situation 2 can be obtained from Situation 1 through the duality transformation $\vec{E}^{inc} \rightarrow \vec{H}^{inc}$, $\vec{H}^{inc} \rightarrow -\vec{E}^{inc}$. Therefore, for a plane wave, a 90^o rotation is equivalent to undergoing the duality transform.

Because of the rotational symmetry of S, the currents excited on S_i under Situation 1 must appear on S_{i+1} under Situation 2. Therefore:

$$K_{i+1,x}^{(2)}(u,v) = -K_{i,y}^{(1)}(u,v)$$
$$K_{i+1,y}^{(2)}(u,v) = K_{i,x}^{(1)}(u,v) \qquad (3\text{-}8)$$
$$K_{i+1,z}^{(2)}(u,v) = K_{i,z}^{(1)}(u,v)$$

$$L_{i+1,x}^{(2)}(u,v) = -L_{i,y}^{(1)}(u,v)$$
$$L_{i+1,y}^{(2)}(u,v) = L_{i,x}^{(1)}(u,v) \qquad (3\text{-}9)$$
$$L_{i+1,z}^{(2)}(u,v) = L_{i,z}^{(1)}(u,v)$$

Assume that Z on S is invertible, the sum currents on S are determined by Eq. (2-22). The tangential components of the incident fields which appear on the right-hand-side of that equation under these two situations are:

$$\begin{bmatrix} \vec{E}_{\tan}^{inc,(1)} \\ \vec{H}_{\tan}^{inc,(1)} \end{bmatrix} = \begin{bmatrix} \hat{u}\cdot\hat{x} \\ \hat{v}\cdot\hat{x} \\ -\hat{u}\cdot\hat{y} \\ -\hat{v}\cdot\hat{y} \end{bmatrix} e^{-ikz} \qquad (3\text{-}10)$$

$$\begin{bmatrix} \vec{E}_{\tan}^{inc,(2)} \\ \vec{H}_{\tan}^{inc,(2)} \end{bmatrix} = \begin{bmatrix} \hat{u}\cdot\hat{y} \\ \hat{v}\cdot\hat{y} \\ \hat{u}\cdot\hat{x} \\ \hat{v}\cdot\hat{x} \end{bmatrix} e^{-ikz} = \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix} \begin{bmatrix} \vec{E}_{\tan}^{inc,(1)} \\ \vec{H}_{\tan}^{inc,(1)} \end{bmatrix} \qquad (3\text{-}11)$$

where I is the 2×2 identity matrix. Therefore,

$$\begin{bmatrix} M & -N \\ N & M \end{bmatrix} \begin{bmatrix} \vec{K}^{(1)} \\ \vec{L}^{(1)} \end{bmatrix} + R \begin{bmatrix} \vec{K}^{(1)} \\ \vec{L}^{(1)} \end{bmatrix} = 2 \begin{bmatrix} \vec{E}_{\tan}^{inc,(1)} \\ \vec{H}_{\tan}^{inc,(1)} \end{bmatrix} \tag{3-12}$$

$$\begin{bmatrix} M & -N \\ N & M \end{bmatrix} \begin{bmatrix} \vec{K}^{(2)} \\ \vec{L}^{(2)} \end{bmatrix} + R \begin{bmatrix} \vec{K}^{(2)} \\ \vec{L}^{(2)} \end{bmatrix} = 2 \begin{bmatrix} \vec{E}_{\tan}^{inc,(2)} \\ \vec{H}_{\tan}^{inc,(2)} \end{bmatrix} = 2 \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix} \begin{bmatrix} \vec{E}_{\tan}^{inc,(1)} \\ \vec{H}_{\tan}^{inc,(1)} \end{bmatrix} \tag{3-13}$$

Since

$$\begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix} \begin{bmatrix} M & -N \\ N & M \end{bmatrix} = \begin{bmatrix} M & -N \\ N & M \end{bmatrix} \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix} \tag{3-14}$$

it follows that if

$$\begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix} R = R \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix} \tag{3-15}$$

we can multiply $\begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix}$ to Eq. (3-12) to get:

$$\begin{bmatrix} M & -N \\ N & M \end{bmatrix} \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix} \begin{bmatrix} \vec{K}^{(1)} \\ \vec{L}^{(1)} \end{bmatrix} + R \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix} \begin{bmatrix} \vec{K}^{(1)} \\ \vec{L}^{(1)} \end{bmatrix} = 2 \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix} \begin{bmatrix} \vec{E}_{\tan}^{inc,(1)} \\ \vec{H}_{\tan}^{inc,(1)} \end{bmatrix} \tag{3-16}$$

Therefore the excited surface currents in these two situations are related by:

$$\begin{bmatrix} \vec{K}^{(2)} \\ \vec{L}^{(2)} \end{bmatrix} = \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix} \begin{bmatrix} \vec{K}^{(1)} \\ \vec{L}^{(1)} \end{bmatrix} = \begin{bmatrix} -\vec{L}^{(1)} \\ \vec{K}^{(1)} \end{bmatrix} \tag{3-17}$$

Combining this result with Eqs. (3-8) and (3-9), we have:

$$\begin{aligned}
K^{(2)}_{i+1,x}(u,v) &= -\,L^{(1)}_{i+1,x}(u,v) = -\,K^{(1)}_{i,y}(u,v) \\
K^{(2)}_{i+1,y}(u,v) &= -\,L^{(1)}_{i+1,y}(u,v) = K^{(1)}_{i,x}(u,v) \\
L^{(2)}_{i+1,x}(u,v) &= K^{(1)}_{i+1,x}(u,v) = -\,L^{(1)}_{i,y}(u,v) \\
L^{(2)}_{i+1,y}(u,v) &= K^{(1)}_{i+1,y}(u,v) = L^{(1)}_{i,x}(u,v)
\end{aligned} \tag{3-18}$$

so that

$$\sum_{i=1}^{4} [K^{(1)}_{i,x}(u,v) + L^{(1)}_{i,y}(u,v)] = 0 \tag{3-19}$$

and

$$\sum_{i=1}^{4} [K^{(1)}_{i,y}(u,v) - L^{(1)}_{i,x}(u,v)] = 0 \tag{3-20}$$

Hence, along the positive z-axis, from Eq (3-5),

$$\begin{aligned}
\vec{E}^{sc}(\vec{r}) &= \frac{ik}{4\pi r}\int\!\!\int_S \Bigg\{ \hat{x}\sum_{i=1}^{4}[K^{(1)}_{i,x}(u,v) + L^{(1)}_{i,y}(u,v)] \\
&\qquad + \hat{y}\sum_{i=1}^{4}\Big[K^{(1)}_{i,y}(u,v) - L^{(1)}_{i,x}(u,v)\Big]\Bigg\} e^{ik\sqrt{(z-z_o)^2+x_o^2+y_o^2}}\,du\,dv \\
&= 0
\end{aligned} \tag{3-21}$$

and the backscattering from S along the positive z-direction must vanish if Eq. (3-15) is satisfied.

C. IMPEDANCE MATRICES FOR ZERO ON-AXIS BACKSCATTERING

It can be verified that the matrix $\begin{bmatrix} Z-\Delta Z^{-1}\Delta & -i\Delta Z^{-1}\sigma_2 \\ -i\sigma_2 Z^{-1}\Delta & \sigma_2 Z^{-1}\sigma_2 \end{bmatrix}$ commutes with $\begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix}$

if and only if:

$$\sigma_2 Z^{-1}\Delta = -\Delta Z^{-1}\sigma_2 \tag{3-22}$$

$$Z - \sigma_2 Z^{-1}\sigma_2 = \Delta Z^{-1}\Delta \tag{3-23}$$

where both Z and Δ are 2×2 matrices. Under the assumption that the inverse of Z exists, we analyze Eqs. (3-22) and (3-23) as follows:

Because of the identity:

$$Z^{-1} = \frac{1}{\det Z}\,\sigma_2 Z^T \sigma_2 \tag{3-24}$$

and, by multiplying σ_2 to both sides of Eq. (3-22):

$$Z^{-1}\Delta = -\sigma_2\,\Delta\,Z^{-1}\sigma_2 \tag{3-25}$$

Eq. (3-23) can now be transformed to

$$\begin{aligned} Z &= -\Delta(\sigma_2 \Delta Z^{-1}\sigma_2) + \sigma_2 Z^{-1}\sigma_2 \\ &= -\Delta\sigma_2\Delta\sigma_2(\sigma_2 Z^{-1}\sigma_2) + \sigma_2 Z^{-1}\sigma_2 \\ &= \frac{1}{\det Z}[1 - (\Delta\sigma_2)^2]Z^T \end{aligned} \tag{3-26}$$

where

$$(\Delta\,\sigma_2)^2 = \Delta\,\sigma_2\,\Delta\,\sigma_2 = \begin{bmatrix} \Delta_{11}\Delta_{22} - \Delta_{12}^2 & 0 \\ 0 & \Delta_{11}\Delta_{22} - \Delta_{21}^2 \end{bmatrix} \tag{3-27}$$

It is observed that Eq. (3-27) is greatly simplified if Δ is symmetric. Therefore, in this thesis, we consider only the case when $\Delta = \Delta^T$. Then $\Delta_{12} = \Delta_{21}$ so that:

$$(\Delta\,\sigma_2)^2 = (\det\Delta)\,I \tag{3-28}$$

From Eq. (3-26),

$$Z = \left(\frac{1 - \det\Delta}{\det Z}\right) Z^T \tag{3-29}$$

Eq. (3-29) can be satisfied only if $\left(\frac{1 - \det\Delta}{\det Z}\right) = \pm 1$. These two cases are:

Case I

$$Z = -Z^T = \begin{bmatrix} 0 & z_{12} \\ -z_{12} & 0 \end{bmatrix}, \; z_{12} \neq 0 \tag{3-30}$$

$$\det\Delta = 1 + \det Z = 1 + z_{12}^2 \tag{3-31}$$

Case II

$$Z = Z^T \tag{3-32}$$

$$\det Z + \det \Delta = 1 \tag{3-33}$$

On the other hand, substituting Eq. (3-24) into Eq. (3-22) yields:

$$z_{11}(\Delta_{12} - \Delta_{21}) = (z_{12} - z_{21})\,\Delta_{11} \tag{3-34}$$

$$z_{22}(\Delta_{12} - \Delta_{21}) = (z_{12} - z_{21})\,\Delta_{22} \tag{3-35}$$

$$z_{11}\Delta_{22} + z_{22}\Delta_{11} = 2z_{21}\Delta_{12} = 2z_{12}\Delta_{21} \tag{3-36}$$

For Case 1, Eqs (3-34) and (3-35) require that $\Delta_{11} = \Delta_{22} = 0$, Eq. (3-36) requires that $\Delta_{12} = \Delta_{21} = 0$. Therefore $\Delta = 0$. From Eq (3-31), $1 + z_{12}^2 = 0$. Therefore $z_{12} = \mp i$ so that $Z^+ = Z^- = Z = \pm\,\sigma_2$.

For Case II, Eqs. (3-34) and (3-35) are trivially satisfied. Eq (3-36) becomes:

$$z_{11}\Delta_{22} + z_{22}\Delta_{11} - 2z_{12}\Delta_{12} = 0 \tag{3-37}$$

Eq. (3-33) is explicitly:

$$z_{11}z_{22} - z_{12}^2 + \Delta_{11}\Delta_{22} - \Delta_{12}^2 = 1 \tag{3-38}$$

The sum of Eq. (3-37) with Eq.(3-38) is:

$$(z_{11} + \Delta_{11})(z_{22} + \Delta_{22}) - (z_{12} + \Delta_{12})^2 = \det(Z + \Delta) = \det Z^+ = 1 \tag{3-39}$$

Subtracting Eq. (3-37) from Eq. (3-38):

$$(z_{11} - \Delta_{11})(z_{22} - \Delta_{22}) - (z_{12} - \Delta_{12})^2 = \det(Z - \Delta) = \det Z^- = 1 \qquad (3\text{-}40)$$

In summary, two sufficient conditions to satisfy Eqs. (3-22) and (3-23) have been deduced under the assumptions that Δ is symmetric and Z is invertible. The first one is:

$$Z^+ = Z^- = \pm\,\sigma_2 \qquad (3\text{-}41)$$

The other, with both Z^+ and Z^- symmetric, is:

$$\det Z^+ = \det Z^- = 1 \qquad (3\text{-}42)$$

It should be noted that if S is a closed surface, the impedance boundary condition closes off the interior of the surface from its exterior. Therefore for the exterior problem, $\det Z^+ = 1$ is sufficient to eliminate the on-axis backscattering from a body of 90° rotational symmetry. Furthermore, Z^+ may vary with location. This is an extension of Weston's theory of isotropic absorbers [4].

IV. SCATTERING OF AN ANISOTROPICALLY COATED CYLINDER

In this chapter, the sum-difference surface current formulation developed in Chapter II is applied to the problem of scattering of an anisotropically coated tubular circular cylinder of finite length and negligible wall thickness. Due to the rotational symmetry, a Fourier series expansion can be utilized to reduce the variables of the problem to only the one along the axis of symmetry, chosen as the z-axis. The Fourier components M_n, N_n of the operators M and N are deduced in terms of the Fourier components of $G(\vec{r}-\vec{r}_0)$ and its partial derivatives. To solve the set of integrodifferential equations so obtained, the equations and the surface currents are both weighted and expanded over the Chebyshev polynomials. The resultant infinite system of linear equations are then truncated and inverted numerically.

A. GEOMETRY AND COORDINATE SYSTEM

Figure 4-1 shows the geometry and coordinate system of a tubular circular cylinder of radius a and length $2h$. The cylindrical coordinate system (ρ, ϕ, z) is scaled so that the surface of the cylinder S is specified by $\rho = 1$ and $-1 \le z \le 1$. The radial vector is thus given by:

$$\begin{aligned} \vec{r} &= a\rho\,\hat{\rho} + hz\hat{z} \\ &= a\rho\,(\cos\phi\hat{x} + \sin\phi\hat{y}) + hz\hat{z} \end{aligned} \tag{4-1}$$

To facilitate representing the tangent vectors over S in matrix form, we chose $\hat{u} = \hat{\phi}$ and $\hat{v} = \hat{z}$ as the orthonormal basis vectors on S, so that $\hat{n} = \hat{\rho}$ is the outward normal to S.

To properly classify the polarization of the incident wave, the rectangular coordinate system (x,y,z) is determined as follows: When a plane wave is not incident along the z-axis, the coordinates are chosen so that the wave is propagating in the zx-plane incident from the half-plane containing the positive x-axis. Axial incidences then are considered as the limiting cases as the direction of incidence approaches the positive or the negative z-axis from this half-plane. Therefore, even for axial incidence, a linearly polarized incident wave having its

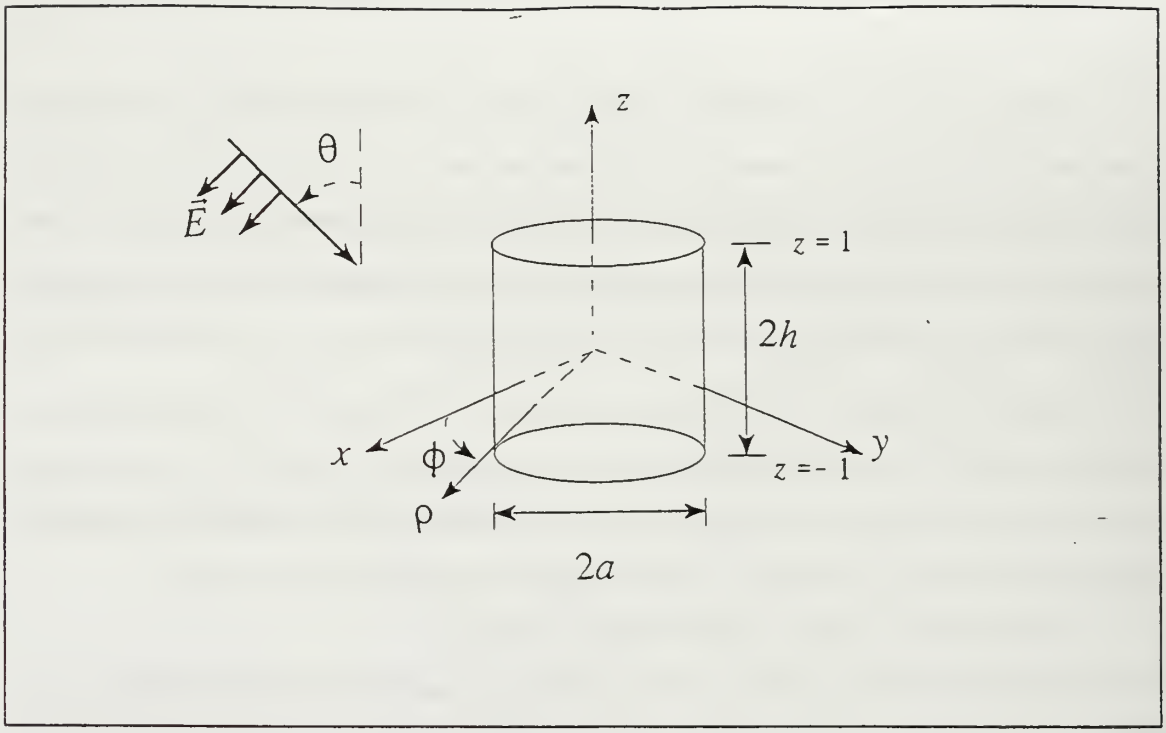

Figure 4-1. The Geometry and Coordinate System.

electric field intensity vector $\vec{E}$ pointing in the y-direction is considered a TE wave while one having its $\vec{E}$ vector in the zx-plane is considered a TM wave. In the spherical coordinate system (r, θ, ϕ), the direction of propagation of the incident wave $\hat{k}$ is given by $\hat{k} = -\hat{z}\cos\theta_i - \hat{x}\sin\theta_i$ where the incident angle θ_i varies over the range $0 \le \theta_i \le \pi$. For an incident plane wave of unit strength, the fields are:

TE - polariation:

$$\vec{E}^{inc} = \hat{y}\ e^{i\vec{k}\cdot\vec{r}}$$
$$\vec{H}^{inc} = \hat{k} \times \vec{E}^{inc} = (\hat{x}\cos\theta_i - \hat{z}\sin\theta_i)\ e^{i\vec{k}\cdot\vec{r}}$$

TM - polarization:

$$\vec{H}^{inc} = \hat{y}\, e^{i\vec{k}\cdot\vec{r}}$$
$$\vec{E}^{inc} = \vec{H}^{inc} \times \hat{k} = -(\hat{x}\cos\theta_i - \hat{z}\sin\theta_i)\, e^{i\vec{k}\cdot\vec{r}}$$

where $\vec{k} = k\hat{k}$. Note that $\vec{k}\bullet\vec{r} = l_1 z\cos\theta_i + l_2\rho\sin\theta_i\cos\phi$ where $l_1 = kh$, and $l_2 = ka$.

On the surfaces S, the tangential components of TE-polarized plane wave are:

$$\begin{aligned} E_\phi^{inc} &= \cos\phi\, e^{i\vec{k}\cdot\vec{r}} \\ H_\phi^{inc} &= -\cos\theta_i\sin\phi\, e^{i\vec{k}\cdot\vec{r}} \\ H_z^{inc} &= -\sin\theta_i\, e^{i\vec{k}\cdot\vec{r}} \end{aligned} \tag{4-1}$$

and the tangential components of TM-polarized plane wave are:

$$\begin{aligned} E_\phi^{inc} &= \cos\theta_i\sin\phi\, e^{i\vec{k}\cdot\vec{r}} \\ E_z^{inc} &= \sin\theta_i\, e^{i\vec{k}\cdot\vec{r}} \\ H_\phi^{inc} &= \cos\phi\, e^{i\vec{k}\cdot\vec{r}} \end{aligned} \tag{4-2}$$

B. SCATTERED FIELDS

From Eq. (2-6), the scattered fields from the cylinder are given in terms of the sum equivalent electric current $\vec{K}$ and the sum magnetic current $\vec{L}$:

$$\begin{aligned} \frac{4\pi}{l_1 l_2}\vec{E}^{sc}(\vec{r}) = &-\frac{1}{k}\nabla\times\int_{-1}^{1}\int_0^{2\pi}\vec{L}(r_o)G(\vec{r}-\vec{r}_o)d\phi_o dz_o \\ &+ i\int_{-1}^{1}\int_0^{2\pi}\vec{K}(r_o)G(\vec{r}-\vec{r}_o)d\phi_o dz_o \\ &- \frac{i}{k^2}\nabla\int_{-1}^{1}\int_0^{2\pi}\vec{K}(r_o)\bullet\nabla_o G(\vec{r}-\vec{r}_o)d\phi_o dz_o \end{aligned} \tag{4-3}$$

and

$$\frac{4\pi}{l_1 l_2}\vec{H}^{sc}(\vec{r}) = \frac{1}{k}\nabla\times\int_{-1}^{1}\int_{0}^{2\pi}\vec{K}(r_o)\ G(\vec{r}-\vec{r}_o)d\phi_o dz_o + i\int_{-1}^{1}\int_{0}^{2\pi}\vec{L}(r_o)\ G(\vec{r}-\vec{r}_o)d\phi_o dz_o - \frac{i}{k^2}\nabla\int_{-1}^{1}\int_{0}^{2\pi}\vec{L}(r_o)\bullet\nabla_o G(\vec{r}-\vec{r}_o)d\phi_o dz_o \tag{4-4}$$

Their components in cylindrical coordinates are:

$$\begin{aligned}\frac{4\pi}{l_1 l_2}E_\rho^{sc}(\vec{r}) = {} & \frac{1}{l_1}\frac{\partial}{\partial z}\int_{-1}^{1}\int_{0}^{2\pi}(\hat{\phi}\bullet\hat{\phi}_o)\,L_\phi(\vec{r}_o)\,G(\vec{r}-\vec{r}_o)\,d\phi_o\,dz_o \\ & - \frac{1}{l_2\rho}\frac{\partial}{\partial\phi}\int_{-1}^{1}\int_{0}^{2\pi}L_z(\vec{r}_o)\,G(\vec{r}-\vec{r}_o)\,d\phi_o\,dz_o \\ & + i\int_{-1}^{1}\int_{0}^{2\pi}(\hat{\rho}\bullet\hat{\phi}_o)\,K_\phi(\vec{r}_o)\,G(\vec{r}-\vec{r}_o)\,d\phi_o\,dz_o \\ & + \frac{i}{l_2^2}\frac{\partial^2}{\partial\rho\partial\phi}\int_{-1}^{1}\int_{0}^{2\pi}K_\phi(\vec{r}_o)\,G(\vec{r}-\vec{r}_o)\,d\phi_o\,dz_o \\ & + \frac{i}{l_1 l_2}\frac{\partial^2}{\partial\rho\partial z}\int_{-1}^{1}\int_{0}^{2\pi}K_z(\vec{r}_o)\,G(\vec{r}-\vec{r}_o)\,d\phi_o\,dz_o\end{aligned} \tag{4-5}$$

$$\begin{aligned}\frac{4\pi}{l_1 l_2}E_\phi^{sc}(\vec{r}) = {} & -\frac{1}{l_1}\frac{\partial}{\partial z}\int_{-1}^{1}\int_{0}^{2\pi}(\hat{\rho}\bullet\hat{\phi}_o)\,L_\phi(\vec{r}_o)\,G(\vec{r}-\vec{r}_o)\,d\phi_o\,dz_o \\ & + \frac{1}{l_2}\frac{\partial}{\partial\rho}\int_{-1}^{1}\int_{0}^{2\pi}L_z(\vec{r}_o)\,G(\vec{r}-\vec{r}_o)\,d\phi_o\,dz_o \\ & + i\int_{-1}^{1}\int_{0}^{2\pi}(\hat{\phi}\bullet\hat{\phi}_o)\,K_\phi(\vec{r}_o)\,G(\vec{r}-\vec{r}_o)\,d\phi_o\,dz_o \\ & + \frac{i}{l_2^2\rho}\frac{\partial^2}{\partial\phi^2}\int_{-1}^{1}\int_{0}^{2\pi}K_\phi(\vec{r}_o)\,G(\vec{r}-\vec{r}_o)\,d\phi_o\,dz_o \\ & + \frac{i}{l_1 l_2\rho}\frac{\partial^2}{\partial\phi\partial z}\int_{-1}^{1}\int_{0}^{2\pi}K_z(\vec{r}_o)\,G(\vec{r}-\vec{r}_o)\,d\phi_o\,dz_o\end{aligned} \tag{4-6}$$

$$
\begin{aligned}
\frac{4\pi}{l_1 l_2} E_z^{sc}(\vec{r}) = &- \frac{1}{l_2\rho}\frac{\partial}{\partial\rho}\int_{-1}^{1}\int_0^{2\pi}(\hat{\phi}\bullet\hat{\phi}_o) L_\phi(\vec{r}_o)\, G(\vec{r}-\vec{r}_o)\, d\phi_o\, dz_o \\
&+ \frac{1}{l_2\rho}\frac{\partial}{\partial\phi}\int_{-1}^{1}\int_0^{2\pi}(\hat{\rho}\bullet\hat{\phi}_o)\, L_\phi(\vec{r}_o)\, G(\vec{r}-\vec{r}_o)\, d\phi_o\, dz_o \\
&+ \frac{i}{l_1 l_2}\frac{\partial^2}{\partial z\partial\phi}\int_{-1}^{1}\int_0^{2\pi} K_\phi(\vec{r}_o)\, G(\vec{r}-\vec{r}_o)\, d\phi_o\, dz_o \\
&+ i\left(1+\frac{1}{l_1^2}\frac{\partial^2}{\partial z^2}\right)\int_{-1}^{1}\int_0^{2\pi} K_z(\vec{r}_o)\, G(\vec{r}-\vec{r}_o)\, d\phi_o\, dz_o
\end{aligned}
\tag{4-7}
$$

$$
\begin{aligned}
\frac{4\pi}{l_1 l_2} H_\rho^{sc}(\vec{r}) = &- \frac{1}{l_1}\frac{\partial}{\partial z}\int_{-1}^{1}\int_0^{2\pi}(\hat{\phi}\bullet\hat{\phi}_o)\, K_\phi(\vec{r}_o)\, G(\vec{r}-\vec{r}_o)\, d\phi_o\, dz_o \\
&+ \frac{1}{l_2\rho}\frac{\partial}{\partial\phi}\int_{-1}^{1}\int_0^{2\pi} K_z(\vec{r}_o)\, G(\vec{r}-\vec{r}_o)\, d\phi_o\, dz_o \\
&+ i\int_{-1}^{1}\int_0^{2\pi}(\hat{\rho}\bullet\hat{\phi}_o)\, L_\phi(\vec{r}_o)\, G(\vec{r}-\vec{r}_o)\, d\phi_o\, dz_o \\
&+ \frac{i}{l_2^2}\frac{\partial^2}{\partial\rho\partial\phi}\int_{-1}^{1}\int_0^{2\pi} L_\phi(\vec{r}_o)\, G(\vec{r}-\vec{r}_o)\, d\phi_o\, dz_o \\
&+ \frac{i}{l_1 l_2}\frac{\partial^2}{\partial\rho\partial z}\int_{-1}^{1}\int_0^{2\pi} L_z(\vec{r}_o)\, G(\vec{r}-\vec{r}_o)\, d\phi_o\, dz_o
\end{aligned}
\tag{4-8}
$$

$$
\begin{aligned}
\frac{4\pi}{l_1 l_2} H_\phi^{sc}(\vec{r}) = &\ \frac{1}{l_1}\frac{\partial}{\partial z}\int_{-1}^{1}\int_0^{2\pi}(\hat{\rho}\bullet\hat{\phi}_o)\, K_\phi(\vec{r}_o)\, G(\vec{r}-\vec{r}_o)\, d\phi_o\, dz_o \\
&- \frac{1}{l_2}\frac{\partial}{\partial\rho}\int_{-1}^{1}\int_0^{2\pi} K_z(\vec{r}_o)\, G(\vec{r}-\vec{r}_o)\, d\phi_o\, dz_o \\
&+ i\int_{-1}^{1}\int_0^{2\pi}(\hat{\phi}\bullet\hat{\phi}_o)\, L_\phi(\vec{r}_o)\, G(\vec{r}-\vec{r}_o)\, d\phi_o\, dz_o \\
&+ \frac{i}{l_2^2\rho}\frac{\partial^2}{\partial\phi^2}\int_{-1}^{1}\int_0^{2\pi} L_\phi(\vec{r}_o)\, G(\vec{r}-\vec{r}_o)\, d\phi_o\, dz_o \\
&+ \frac{i}{l_1 l_2\rho}\frac{\partial^2}{\partial\phi\partial z}\int_{-1}^{1}\int_0^{2\pi} L_z(\vec{r}_o)\, G(\vec{r}-\vec{r}_o)\, d\phi_o\, dz_o
\end{aligned}
\tag{4-9}
$$

$$\frac{4\pi}{l_1 l_2} H_z^{sc}(\vec{r}) = \frac{1}{l_2\rho}\frac{\partial}{\partial\rho}\int_{-1}^{1}\int_0^{2\pi} \rho\,(\hat{\phi}\bullet\hat{\phi}_o)\, K_\phi(\vec{r}_o)\, G(\vec{r}-\vec{r}_o)\, d\phi_o\, dz_o$$
$$- \frac{1}{l_2\rho}\frac{\partial}{\partial\phi}\int_{-1}^{1}\int_0^{2\pi} (\hat{\rho}\bullet\hat{\phi}_o)\, K_\phi(\vec{r}_o)\, G(\vec{r}-\vec{r}_o)\, d\phi_o\, dz_o$$
$$+ \frac{i}{l_1 l_2}\frac{\partial^2}{\partial z\partial\phi}\int_{-1}^{1}\int_0^{2\pi} L_\phi(\vec{r}_o)\, G(\vec{r}-\vec{r}_o)\, d\phi_o\, dz_o \tag{4-10}$$
$$+ i\left(1 + \frac{1}{l_1^2}\frac{\partial^2}{\partial z^2}\right)\int_{-1}^{1}\int_0^{2\pi} L_z(\vec{r}_o)\, G(\vec{r}-\vec{r}_o)\, d\phi_o\, dz_o$$

Note that in the above equations,

$$\hat{\rho} = \hat{\rho}_o\cos(\phi-\phi_o) + \hat{\phi}_o\sin(\phi-\phi_o) \tag{4-11}$$

$$\hat{\phi} = -\hat{\rho}_o\sin(\phi-\phi_o) + \hat{\phi}_o\cos(\phi-\phi_o) \tag{4-12}$$

Because of the rotational symmetry $G(\vec{r}-\vec{r}_o)$ depends on $\phi-\phi_o$ and Fourier series can be introduced to eliminate the variable ϕ. Define the Fourier series expansion of a function $f(\phi,z)$ by:

$$f(\phi,z) = \sum_{n=-\infty}^{\infty} e^{in\phi} f_n(z) \tag{4-13}$$

then

$$G(\vec{r}-\vec{r}_o) = \frac{e^{ik|\vec{r}-\vec{r}_o|}}{k|\vec{r}-\vec{r}_o|} = \sum_{n=\infty}^{\infty} e^{in(\phi-\phi_o)} G_n(l_1|z-z_o|, l_2, \rho) \tag{4-14}$$

and

$$G_n(l_1|z-z_o|, l_2, \rho) = \int_{-\pi}^{\pi} \frac{d\phi}{2\pi} e^{-in(\phi-\phi_o)} G(\vec{r}-\vec{r}_o) \tag{4-15}$$

Eqs. (4-5) to (4-10) become:

$$\begin{aligned} \frac{2}{l_1 l_2} E_{\rho n}^{sc}(\rho,z) = & \frac{1}{2l_1} \frac{\partial}{\partial z} \int_{-1}^{1} L_{\phi n}(z_o)[G_{n-1}+G_{n+1}]dz_o \\ & - \frac{in}{l_2\rho} \int_{-1}^{1} L_{zn}(z_o) G_n dz_o \\ & + \frac{1}{2} \int_{-1}^{1} K_{\phi n}(z_o)\ [G_{n-1}-G_{n+1}]dz_o - \frac{n}{l_2^2} \frac{\partial}{\partial \rho} \int_{-1}^{1} K_{\phi n}(z_o)\ G_n dz_o \\ & + \frac{i}{l_1 l_2} \frac{\partial}{\partial \rho} \int_{-1}^{1} \left[\frac{\partial}{\partial z_o} K_{zn}(z_o)\right] G_n dz_o \end{aligned} \tag{4-16}$$

$$\begin{aligned} \frac{2}{l_1 l_2} E_{\phi n}^{sc}(\rho,z) = & \frac{i}{2l_1} \frac{\partial}{\partial z} \int_{-1}^{1} L_{\phi n}(z_o)[G_{n-1}-G_{n+1}]dz_o + \frac{1}{l_2} \frac{\partial}{\partial \rho} \int_{-1}^{1} L_{zn}(z_o) G_n dz_o \\ & + i \int_{-1}^{1} K_{\phi n}(z_o) \left[\frac{1}{2}(G_{n-1}+G_{n+1}) - \frac{n^2}{l_2^2 \rho} G_n\right] dz_o \\ & - \frac{n}{l_1 l_2 \rho} \int_{-1}^{1} \left[\frac{\partial}{\partial z_o} K_{zn}(z_o)\right] G_n dz_o \end{aligned} \tag{4-17}$$

$$\begin{aligned} \frac{2}{l_1 l_2} E_{zn}^{sc}(\rho,z) = & \frac{1}{2l_2\rho} \int_{-1}^{1} L_{\phi n}(z_o)[(n-1)G_{n-1}-(n+1)G_{n+1}]dz_o \\ & - \frac{1}{2l_2} \frac{\partial}{\partial \rho} \int_{-1}^{1} L_{\phi n}(z_o)[G_{n-1}+G_{n+1}]dz_o \\ & - \frac{n}{l_1 l_2} \frac{\partial}{\partial z} \int_{-1}^{1} K_{\phi n}(z_o)\ G_n dz_o \\ & + \frac{i}{l_1^2} \frac{\partial}{\partial z} \int_{-1}^{1} \left[\frac{\partial}{\partial z_o} K_{zn}(z_o)\right] G_n dz_o + i \int_{-1}^{1} K_{zn}(z_o)\ G_n dz_o \end{aligned} \tag{4-18}$$

$$
\begin{aligned}
\frac{2}{l_1 l_2} H^{sc}_{\rho n}(\rho,z) = & -\frac{1}{2l_1}\frac{\partial}{\partial z}\int_{-1}^{1} K_{\phi n}(z_o)[G_{n-1}+G_{n+1}]dz_o \\
& + \frac{in}{l_2\rho}\int_{-1}^{1} K_{zn}(z_o)G_n dz_o \\
& + \frac{1}{2}\int_{-1}^{1} L_{\phi n}(z_o)\,[G_{n-1}-G_{n+1}]dz_o - \frac{n}{l_2^2}\frac{\partial}{\partial\rho}\int_{-1}^{1} L_{\phi n}(z_o)\;G_n dz_o \\
& + \frac{i}{l_1 l_2}\frac{\partial}{\partial\rho}\int_{-1}^{1}\left[\frac{\partial}{\partial z_o}L_{zn}(z_o)\right] G_n dz_o
\end{aligned}
\tag{4-19}
$$

$$
\begin{aligned}
\frac{2}{l_1 l_2} H^{sc}_{\phi n}(\rho,z) = & -\frac{i}{2l_1}\frac{\partial}{\partial z}\int_{-1}^{1} K_{\phi n}(z_o)[G_{n-1}-G_{n+1}]dz_o - \frac{1}{l_2}\frac{\partial}{\partial\rho}\int_{-1}^{1} K_{zn}(z_o)G_n dz_o \\
& + i\int_{-1}^{1} L_{\phi n}(z_o)\,[\frac{1}{2}(G_{n-1}+G_{n+1}) - \frac{n^2}{l_2^2\rho}G_n]dz_o \\
& - \frac{n}{l_1 l_2\rho}\int_{-1}^{1}[\frac{\partial}{\partial z}L_{zn}(z_o)]\;G_n dz_o
\end{aligned}
\tag{4-20}
$$

$$
\begin{aligned}
\frac{2}{l_1 l_2} H^{sc}_{zn}(\rho,z) = & -\frac{1}{2l_2\rho}\int_{-1}^{1} K_{\phi n}(z_o)[(n-1)G_{n-1}-(n+1)G_{n+1}]dz_o \\
& + \frac{1}{2l_2}\frac{\partial}{\partial\rho}\int_{-1}^{1} K_{\phi n}(z_o)[G_{n-1}+G_{n+1}]dz_o \\
& - \frac{n}{l_1 l_2}\frac{\partial}{\partial z}\int_{-1}^{1} L_{\phi n}(z_o)\;G_n dz_o \\
& + \frac{i}{l_1^2}\frac{\partial}{\partial z}\int_{-1}^{1}[\frac{\partial}{\partial z_o}L_{zn}(z_o)]\;G_n dz_o + i\int_{-1}^{1} L_{zn}(z_o)\;G_n dz_o
\end{aligned}
\tag{4-21}
$$

Note that in Eqs. (4-16) to (4-21), G_n stands for $G_n(l_1|z-z_o|, l_2, \rho)$. In shifting the z-derivative to the current densities $K_{zn}(z_0)$ and $L_{zn}(z_0)$, the fact that edge conditions [5] require that these components of the scattering currents vanish as $|z_0|$ approach 1 from below is used.

As $\rho \to 1^{\pm}$, the operators M_n and N_n can be deduced from Eqs. (4-16) to (4-21):

$$
\begin{aligned}
M_{n,11} &= -\, i\, l_1 l_2 \int_{-1}^{1} dz_o \left[\frac{1}{2}(G_{n-1} + G_{n+1}) - \frac{n^2}{l_2^2} G_n\right]_{\rho=1} \\
M_{n,12} &= n \int_{-1}^{1} dz_o \, [G_n]_{\rho=1} \frac{\partial}{\partial z_o} \\
M_{n,21} &= n \frac{\partial}{\partial z} \int_{-1}^{1} dz_o \, [G_n]_{\rho=1} \\
M_{n,22} &= -\, i l_1 l_2 \int_{-1}^{1} dz_o \left[[G_n]_{\rho=1} + \frac{1}{l_1^2} \frac{\partial}{\partial z} [G_n]_{\rho=1} \frac{\partial}{\partial z_o}\right]
\end{aligned}
\tag{4-22}
$$

$$
\begin{aligned}
N_{n,11} &= \frac{i}{2} l_2 \frac{\partial}{\partial z} \int_{-1}^{1} dz_o \, [G_{n-1} - G_{n+1}]_{\rho=1} \\
N_{n,12} &= \frac{1}{2} l_1 l_2 \int_{-1}^{1} dz_o \, \frac{\partial}{\partial l_2} [G_n]_{\rho=1} \\
N_{n,21} &= \frac{l_1}{2} \int_{-1}^{1} dz_o \left\{ [(n-1) G_{n-1} - (n+1) G_{n+1}]_{\rho=1} - \frac{1}{2} l_2 \frac{\partial}{\partial l_2} [G_{n-1} + G_{n+1}]_{\rho=1} \right\} \\
N_{n,22} &= 0
\end{aligned}
\tag{4-23}
$$

Note that:

$$
\frac{i}{l_2}\left(\frac{\partial}{\partial \rho}\Big|_{\rho=1^-} + \frac{\partial}{\partial \rho}\Big|_{\rho=1^+} \right) G_n(l_1 |z - z_0|, l_2, \rho) = \frac{\partial}{\partial l_2} G_n(l_1 |z - z_0|, l_2, 1) \tag{4-24}
$$

Assuming that Z is invertible, from Eq. (2-23), the equations for the sum currents are therefore:

$$
\begin{bmatrix} M_n & -N_n \\ N_n & M_n \end{bmatrix} \begin{bmatrix} \vec{K}_n \\ \vec{L}_n \end{bmatrix} + R \begin{bmatrix} \vec{K}_n \\ \vec{L}_n \end{bmatrix} = 2 \begin{bmatrix} \vec{E}^{inc}_{\tan,n} \\ \vec{H}^{inc}_{\tan,n} \end{bmatrix} \tag{4-25}
$$

where

$$R = \begin{bmatrix} Z - \Delta Z^{-1}\Delta & -i\Delta Z^{-1}\sigma_2 \\ -i\sigma_2 Z^{-1}\Delta & \sigma_2 Z^{-1}\sigma_2 \end{bmatrix} \tag{4-26}$$

and the vectors $\vec{K}_n$, $\vec{L}_n$, $\vec{E}^{inc}_{\tan,n}$, $\vec{L}^{inc}_{\tan,n}$ are two dimensional column matrix representations of the respective tangential vector fields on S over the orthonomal basis $\hat{\phi}$, $\hat{z}$. The incident fields are, for TE and TM polarizations respectively:

TE:

$$\begin{aligned} E^{inc}_{\phi,n} &= (-i)^{n-1} J_n^{\prime}(l_2 \sin\theta_i)\, e^{-il_1 z\cos\theta_i} \\ H^{inc}_{\phi,n} &= -\frac{n(-i)^n \cos\theta_i}{l_2 \sin\theta_i} J_n(l_2 \sin\theta_i)\, e^{-il_1 z\cos\theta_i} \\ H^{inc}_{z,n} &= -\sin\theta_i (-i)^{n-1} J_n(l_2 \sin\theta_i)\, e^{-il_1 z\cos\theta_i} \end{aligned} \tag{4-27}$$

TM:

$$\begin{aligned} E^{inc}_{\phi,n} &= \frac{n(-i)^n \cos\theta_i}{l_2 \sin\theta_i} J_n(l_2 \sin\theta_i)\, e^{-il_1 z\cos\theta_i} \\ E^{inc}_{z,n} &= \sin\theta_i (-i)^n J_n(l_2 \sin\theta_i)\, e^{-il_1 z\cos\theta_i} \\ H^{inc}_{\phi,n} &= (-i)^{n-1} J_n^{\prime}(l_2 \sin\theta_i)\, e^{-il_1 z\cos\theta_i} \end{aligned} \tag{4-28}$$

C. TRANSFORM TO SYSTEM OF LINEAR EQUATIONS

Since the surface current components $K_\phi(\phi, z_o) = O(1 - z_o^2)^{-1/2}$ and $K_z(\phi, z_o) = O(1 - z_o^2)^{1/2}$ as $|z_o| \to 1^-$, representations of $K_{\phi n}(z_o)$ and $\frac{d}{dz_o} K_{zn}(z_o)$ in Chebyshev polynomials of the first kind combined with the weighting factor conform to the proper edge behavior of the currents:

$$
\begin{aligned}
K_{\phi,n}(z_o) &= \frac{1}{\pi \sin v} \sum_{q=0}^{\infty} K_{\phi,n}^{q} \, T_p(z_o) \\
&= \frac{1}{\pi \sin v} \sum_{q=0}^{\infty} K_{\phi,n}^{q} \cos qv \\
K_{z,n}(z_o) &= \frac{1}{\pi} \sum_{q=0}^{\infty} K_{z,n}^{q} \sin(q+1)v
\end{aligned}
\tag{4-29}
$$

where $T_p(z_o)$ is the Chebyshev polynomials of the first kind and $z_0 = \cos v$, $-1 \le z_0 \le 1$. Similarly,

$$
\begin{aligned}
L_{\phi,n}(z_o) &= \frac{1}{\pi \sin v} \sum_{q=0}^{\infty} L_{\phi,n}^{q} \cos qv \\
L_{z,n}(z_o) &= \frac{1}{\pi} \sum_{q=0}^{\infty} L_{z,n}^{q} \sin(q+1)v
\end{aligned}
\tag{4-30}
$$

For an invertible Z, the coefficients in the above equations are to be determined from Eq. (4-23). To make use of the orthogonal property of the Chebyshev polynomials, the factor $\sin v \sin(p+1)v$ for $p \ge 0$ is multiplied to both sides of Eq. (4-23) before an integration over the range of v is carried out. The results are described term by term in the subsection to follow. This procedure creates an infinite system of linear equations to be solved numerically after it is truncated at an appropriate order determined by the electrical size of the cylinder.

1. Incident Fields

$$
\begin{bmatrix} \vec{E}_{\tan,n}^{inc,p} \\ \vec{H}_{\tan,n}^{inc,p} \end{bmatrix} = \frac{2}{p+1} \int_0^{\pi} \frac{dv}{\pi} \sin v \, \sin(p+1)v \begin{bmatrix} \vec{E}_{\tan,n}^{inc} \\ \vec{H}_{\tan,n}^{inc} \end{bmatrix}
\tag{4-31}
$$

TE:

$$
\begin{aligned}
E_{\phi,n}^{inc,p} &= (-i)^{n+p-1}\left(J_{p-1}(l_1\cos\theta_i) + J_{p+1}(l_1\cos\theta_i)\right) J_n^{\prime}(l_2\sin\theta_i) \\
H_{\phi,n}^{inc,p} &= -\frac{2n(-i)^{n+p}}{l_1 l_2\sin\theta_i} J_{p+1}(l_1\cos\theta_i)\, J_n(l_2\sin\theta_i) \\
H_{z,n}^{inc,p} &= -(-i)^{n+p}\sin\theta_i\left(J_{p-1}(l_1\cos\theta_i) + J_{p+1}(l_1\cos\theta_i)\right) J_n(l_2\sin\theta_i)
\end{aligned}
\tag{4-32}
$$

TM:

$$
\begin{aligned}
E_{\phi,n}^{inc,p} &= \frac{2n(-i)^{n+p}}{l_1 l_2\sin\theta_i} J_{p+1}(l_1\cos\theta_i)\, J_n(l_2\sin\theta_i) \\
E_{z,n}^{inc,p} &= (-i)^{n+p}\sin\theta_i\left(J_{p-1}(l_1\cos\theta_i) + J_{p+1}(l_1\cos\theta_i)\right) J_n(l_2\sin\theta_i) \\
H_{\phi,n}^{inc,p} &= (-i)^{n+p-1}\left(J_{p-1}(l_1\cos\theta_i) + J_{p+1}(l_1\cos\theta_i)\right) J_n^{\prime}(l_2\sin\theta_i)
\end{aligned}
\tag{4-33}
$$

where $J^{\prime}(\bullet)$ is a derivative with respect to the argument. Note that as θ_i approaches 0 or π, only $n = \pm 1$ terms are nonzero. Hence only $n = \pm 1$ currents exist. For axial incident when $\theta_i = 0$:

TE:

$$
\begin{aligned}
E_{\phi,1}^{inc,p} &= \frac{(-i)^p}{l_1} J_{p+1}(l_1) = E_{\phi,-1}^{inc,p} \\
H_{\phi,1}^{inc,p} &= -\frac{(-i)^{p+1}}{l_1} J_{p+1}(l_1) = -H_{\phi,-1}^{inc,p}
\end{aligned}
\tag{4-34}
$$

TM:

$$
\begin{aligned}
E_{\phi,1}^{inc,p} &= \frac{(-i)^{p+1}}{l_1} J_{p+1}(l_1) = -E_{\phi,-1}^{inc,p} \\
H_{\phi,1}^{inc,p} &= \frac{(-i)^p}{l_1} J_{p+1}(l_1) = H_{\phi,-1}^{inc,p}
\end{aligned}
\tag{4-35}
$$

2. The R - Matrix Term

$$
\begin{aligned}
&\frac{2}{p+1}\int_0^{\pi}\frac{dv}{\pi}\ \sin v\ \sin(p+1)vK_{\phi n}(\cos v) \\
&= \frac{2}{(p+1)\pi^2}\sum_{q=0}^{\infty} K_{\phi,n}^{q}\int_0^{\pi} dv\ \sin(p+1)v\ \cos qv \\
&= \sum_{q=0}^{\infty} A_{\phi}^{p,q}\ K_{\phi,n}^{q}
\end{aligned}
\tag{4-36}
$$

where

$$
A_{\phi}^{p,q} = \begin{cases} 0 & p+q \quad \text{odd} \\ \dfrac{4}{\pi^2(p+q+1)\,(p-q+1)} & p+q \quad \text{even} \end{cases}
\tag{4-37}
$$

$$
\begin{aligned}
&\frac{2}{p+1}\int_0^{\pi}\frac{dv}{\pi}\ \sin v\ \sin(p+1)v\ K_{zn}(\cos v) \\
&= \frac{2}{(p+1)\pi^2}\sum_{q=0}^{\infty} K_{z,n}^{q}\int_0^{\pi} dv\ \sin v\ \sin(p+1)v\ \sin(q+1)v \\
&= \sum_{q=0}^{\infty} A_{z}^{p,q}\ K_{z,n}^{q}
\end{aligned}
\tag{4-38}
$$

where

$$
A_{z}^{p,q} = \begin{cases} 0 & p+q \quad \text{odd} \\ \dfrac{-8\,(q+1)}{\pi^2(p+q+1)\,(p-q+1)\,(p+q+3)\,(p-q-1)} & p+q \quad \text{even} \end{cases}
\tag{4-39}
$$

Therefore:

$$\frac{2}{p+1}\int_0^{\pi}\frac{dv}{\pi}\ \sin v\ \sin(p+1)v\ R\begin{bmatrix}\vec{K}_n\\ \vec{L}_n\end{bmatrix} = \sum_{q=0}^{\infty} R\begin{bmatrix}A_\phi^{p,q} & 0 & 0 & 0\\ 0 & A_z^{p,q} & 0 & 0\\ 0 & 0 & A_\phi^{p,q} & 0\\ 0 & 0 & 0 & A_z^{p,q}\end{bmatrix}\begin{bmatrix}\vec{K}_n^q\\ \vec{L}_n^q\end{bmatrix} \tag{4-40}$$

$$= \sum_{q=0}^{\infty} R^{p,q}\begin{bmatrix}\vec{K}_n^q\\ \vec{L}_n^q\end{bmatrix}$$

where $R^{p,q}$ is a four-by-four matrix, given in terms of R by

$$\left.\begin{aligned} R_{i1}^{p,q} &= R_{i1}\ A_\phi^{p,q}\\ R_{i2}^{p,q} &= R_{i2}\ A_z^{p,q}\\ R_{i3}^{p,q} &= R_{i3}\ A_\phi^{p,q}\\ R_{i4}^{p,q} &= R_{i4}\ A_z^{p,q}\end{aligned}\right\}\quad \text{for}\quad i = 1,\ 2,\ 3,\ 4 \tag{4-41}$$

Note that $R_{ij}^{p,q} = 0$ if $p+q$ is odd.

3. The M_n, N_n Operators

Define the 4x4 matrix $X_n^{p,q}$ by:

$$\frac{2}{p+1}\int_0^{\pi}\frac{dv}{\pi}\ \sin v\sin(p+1)v\begin{bmatrix}M_n & -N_n\\ N_n & M_n\end{bmatrix}\begin{bmatrix}\vec{K}_n\\ \vec{L}_n\end{bmatrix} = \sum_{q=0}^{\infty} X_n^{p,q}\begin{bmatrix}\vec{K}_n^q\\ \vec{L}_n^q\end{bmatrix} \tag{4-42}$$

$$\frac{2}{p+1}\int_0^{\pi}\frac{dv}{\pi}\ \sin v\ \sin(p+1)v\ M_{n,11}\ [K_{\phi,n}(z_o)]$$
$$= \sum_{q=0}^{\infty}\frac{-il_1l_2}{p+1}\int_0^{\pi}\frac{dv}{\pi}[\cos pv\cos(p+2)v]\int_0^{\pi}\frac{dv_o}{\pi}\cos qv_o\left[\frac{1}{2}(G_{n-1}+G_{n+1})-\frac{n^2}{l_2^2}G_n\right]K_{\phi,n}^q$$
$$= \sum_{q=0}^{\infty} X_{n,11}^{p,q}\ K_{\phi,n}^q$$

and,

$$X_{n,11}^{p,q} = -\frac{il_1 l_2}{p+1}\left[\frac{1}{2}(G_{n-1}^{p,q} - G_{n-1}^{p+2,q} + G_{n+1}^{p,q} - G_{n+1}^{p+2,q}) - \frac{n^2}{l_2^2}(G_n^{p,q} - G_n^{p+2,q}\right] \tag{4-43}$$

$$\begin{aligned}
&\frac{2}{p+1}\int_0^\pi \frac{dv}{\pi} \ \sin v \ \sin(p+1)v \ M_{n,21} \ [K_{\phi,n}(z_o)] \\
&= \sum_{q=0}^{\infty} \frac{-2n}{p+1} \int_0^\pi \frac{dv}{\pi} \sin(p+1)v \frac{\partial}{\partial v}\int_0^\pi \frac{dv_o}{\pi} \cos qv_o \, G_n K_{\phi,n}^q \\
&= \sum_{q=0}^{\infty} 2n \int_0^\pi \frac{dv}{\pi} \cos(p+1)v \int_0^\pi \frac{dv_o}{\pi} \cos qv_o \, G_n K_{\phi,n}^q \\
&= \sum_{q=0}^{\infty} X_{n,21}^{p,q} \ K_{\phi,n}^q
\end{aligned}$$

and,

$$X_{n,21}^{p,q} = 2n\, G_n^{p+1,q} \tag{4-44}$$

$$\begin{aligned}
&\frac{2}{p+1}\int_0^\pi \frac{dv}{\pi} \ \sin v \ \sin(p+1)v \ N_{n,11} \ [K_{\phi,n}(z_o)] \\
&= \sum_{q=0}^{\infty} \frac{-il_1}{p+1} \int_0^\pi \frac{dv}{\pi} \sin(p+1)v \frac{\partial}{\partial v}\int_0^\pi \frac{dv_o}{\pi} \cos qv_o [G_{n-1} - G_{n+1}] K_{\phi,n}^q \\
&= \sum_{q=0}^{\infty} X_{n,31}^{p,q} \ K_{\phi,n}^q
\end{aligned}$$

and,

$$X_{n,31}^{p,q} = il_2 [G_{n-1}^{p+1,q} - G_{n+1}^{p+1,q}] \tag{4-45}$$

$$\frac{2}{p+1}\int_0^{\pi}\frac{dv}{\pi}\ \sin v\ \sin(p+1)v\ N_{n,21}\ [K_{\phi,n}(z_o)]$$

$$= \sum_{q=0}^{\infty}\frac{l_1}{p+1}\int_0^{\pi}\frac{dv}{\pi}\sin v\sin(p+1)v\int_0^{\pi}\frac{dv_o}{\pi}\cos qv_o$$

$$\left[(n-1)G_{n-1}-(n+1)G_{n+1}-\frac{l_2}{2}\frac{\partial}{\partial l_2}(G_{n-1}+G_{n+1})\right]K_{\phi,n}^{q}$$

$$= \sum_{q=0}^{\infty}X_{n,41}^{p,q}\ K_{\phi,n}^{q}$$

and,

$$X_{n,41}^{p,q} = \frac{l_1}{2(p+1)}\Big[(n-1)(G_{n-1}^{p,q}-G_{n-1}^{p+2,q})-(n+1)(G_{n+1}^{p,q}-G_{n+1}^{p+2,q}) - \frac{l_2}{2}\ \frac{\partial}{\partial l_2}(G_{n-1}^{p,q}-G_{n-1}^{p+2,q}+G_{n+1}^{p,q}-G_{n+1}^{p+2,q})\Big] \tag{4-46}$$

$$\frac{2}{p+1}\int_0^{\pi}\frac{dv}{\pi}\ \sin v\ \sin(p+1)v\ M_{n,12}\ [K_{z,n}(z_o)]$$

$$= \sum_{q=0}^{\infty}\frac{-2n}{p+1}\int_0^{\pi}\frac{dv}{\pi}\sin v\sin(p+1)v\int_0^{\pi}\frac{dv_o}{\pi}(q+1)\cos(q+1)v_o\,G_n K_{z,n}^{q}$$

$$= \sum_{q=0}^{\infty}X_{n,12}^{p,q}\ K_{z,n}^{q}$$

and,

$$X_{n,12}^{p,q} = -\frac{n\,(q+1)}{p+1}[G_n^{p,q+1}-G_n^{p+2,q+1}] \tag{4-47}$$

$$
\begin{aligned}
&\frac{2}{p+1}\int_0^{\pi}\frac{dv}{\pi}\ \sin v\ \sin(p+1)v\ M_{n,22}\ [K_{z,n}(z_o)] \\
&= \sum_{q=0}^{\infty}\frac{-2il_1l_2}{p+1}\left\{\int_0^{\pi}\frac{dv}{\pi}\sin v\sin(p+1)v\int_0^{\pi}\frac{dv_o}{\pi}\sin v_o\sin(q+1)v_o\,G_n\right. \\
&\qquad\left.+\frac{1}{l_1^2}\int_0^{\pi}\frac{dv}{\pi}\sin(p+1)v\,\frac{\partial}{\partial v}\int_0^{\pi}\frac{dv_o}{\pi}\,(q+1)\cos(q+1)v_o\,G_n\right\}K_{z,n}^{q} \\
&= \sum_{q=0}^{\infty}X_{n,22}^{p,q}\ K_{z,n}^{q}
\end{aligned}
$$

and,

$$
X_{n,22}^{p,q} = -\frac{il_1l_2}{2}\left[\frac{1}{p+1}(G_n^{p,q}+G_n^{p+2,q+2}-G_n^{p,q+2}-G_n^{p+2,q})-\frac{4(q+1)}{l_1^2}G_n^{p+1,q+1}\right] \tag{4-48}
$$

$$
\begin{aligned}
&\frac{2}{p+1}\int_0^{\pi}\frac{dv}{\pi}\ \sin v\ \sin(p+1)v\ N_{n,12}\ [K_{z,n}(z_o)] \\
&= \sum_{q=0}^{\infty}\frac{l_1l_2}{p+1}\int_0^{\pi}\frac{dv}{\pi}\sin v\sin(p+1)v\int_0^{\pi}\frac{dv_o}{\pi}\sin v_o\sin(q+1)v_o\left[\frac{\partial}{\partial l_2}G_n\right]K_{z,n}^{q} \\
&= \sum_{q=0}^{\infty}X_{n,32}^{p,q}\ K_{z,n}^{q}
\end{aligned}
$$

and,

$$
X_{n,32}^{p,q} = \frac{l_1l_2}{4(p+1)}\ \frac{\partial}{\partial l_2}\,[G_n^{p,q}+G_n^{p+2,q+2}-G_n^{p+2,q}-G_n^{p,q+2}] \tag{4-49}
$$

Note that, for *n=0*, the expressions simplify to:

$$
\begin{aligned}
X_{o,11}^{p,q} &= -\frac{il_1 l_2}{p+1}[G_1^{p,q} - G_1^{p+2,q}] \\
X_{o,41}^{p,q} &= -\frac{l_1}{p+1}[(G_1^{p,q} - G_1^{p+2,q}) + \frac{l_2}{2}\frac{\partial}{\partial l_2}(G_1^{p,q} - G_1^{p+2,q})] \\
X_{o,22}^{p,q} &= -\frac{il_1 l_2}{2}\left[\frac{1}{p+1}(G_o^{p,q} + G_o^{p+2,q+2} - G_o^{p,q+2} - G_o^{p+2,q}) - \frac{4(q+1)}{l_1^2}G_o^{p+1,q+1}\right] \\
X_{o,32}^{p,q} &= \frac{l_1 l_2}{4(p+1)}\frac{\partial}{\partial l_2}(G_o^{p,q} + G_o^{p+2,q+2} - G_o^{p,q+2} - G_o^{p+2,q}) \\
X_{o,21}^{p,q} &= X_{o,31}^{p,q} = X_{o,44}^{p,q} = X_{o,12}^{p,q} = 0
\end{aligned}
\tag{4-50}
$$

Also note that $X_o^{p,q} = 0$ if $p+q$ is odd.

From the symmetry of $\begin{bmatrix} M_n & -N_n \\ N_n & M_n \end{bmatrix}$, we can deduce that:

$$
\begin{aligned}
X_{n,13}^{p,q} &= -X_{n,31}^{p,q} & \qquad X_{n,14}^{p,q} &= -X_{n,32}^{p,q} \\
X_{n,23}^{p,q} &= -X_{n,41}^{p,q} & \qquad X_{n,24}^{p,q} &= X_{n,42}^{p,q} = 0 \\
X_{n,33}^{p,q} &= X_{n,11}^{p,q} & \qquad X_{n,34}^{p,q} &= X_{n,12}^{p,q} \\
X_{n,43}^{p,q} &= X_{n,21}^{p,q} & \qquad X_{n,44}^{p,q} &= X_{n,22}^{p,q}
\end{aligned}
\tag{4-51}
$$

Note that for $p+q$ odd,

$$
X_{n,11}^{p,q} = X_{n,41}^{p,q} = X_{n,22}^{p,q} = X_{n,32}^{p,q} = X_{n,23}^{p,q} = X_{n,33}^{p,q} = X_{n,14}^{p,q} = X_{n,44}^{p,q} = 0 \tag{4-52}
$$

and for $p+q$ even,

$$
X_{n,21}^{p,q} = X_{n,31}^{p,q} = X_{n,12}^{p,q} = X_{n,13}^{p,q} = X_{n,43}^{p,q} = X_{n,34}^{p,q} = 0 \tag{4-53}
$$

The integrodifferential Eq. (4-25) is thus transformed into the infinite system of linear equations of the unknown coefficients $\begin{bmatrix} \vec{K}_n^q \\ \vec{L}_n^q \end{bmatrix}$:

$$\sum_{q=0}^{\infty} [X_n^{p,q} + R^{p,q}] \begin{bmatrix} \vec{K}_n^q \\ \vec{L}_n^q \end{bmatrix} = 2 \begin{bmatrix} \vec{E}_{\tan,n}^{inc,p} \\ \vec{H}_{\tan,n}^{inc,p} \end{bmatrix} \tag{4-54}$$

which is to be solved numerically.

D. RADIATION IN THE FAR FIELD

In the far field, we can write Eq. (4-3) as:

$$\begin{aligned} \frac{4\pi}{l_1 l_2} \vec{E}^{sc}(\vec{r}) \approx &+ i\int_{-1}^{1} dz_o \int_{o}^{2\pi} d\phi_o [\vec{L}(\vec{r}_o) \times \hat{r}] G(\vec{r} - \vec{r}_o) \\ &+ i\int_{-1}^{1} dz_o \int_{o}^{2\pi} d\phi_o \Big\{ \vec{K}(\vec{r}_o) - i\hat{r} [K_\phi(\vec{r}_o) \sin\theta \sin(\phi - \phi_o) \\ &+ K_z(\vec{r}_o) \cos\theta] \Big\} G(\vec{r} - \vec{r}_o) \end{aligned} \tag{4-55}$$

or equivalently,

$$\frac{\vec{E}^{sc}(\vec{r})}{G(\vec{r})} = \frac{il_1 l_2}{2}\int_{-1}^{1} dz_o \int_o^{2\pi} \frac{d\phi_o}{2\pi}\Big\{\hat{\theta}\,[K_\phi \cos\theta \sin(\phi-\phi_o) - K_z \sin\theta + L_\phi \cos(\phi-\phi_o)]$$

$$+ \hat{\phi}\,[K_\phi \cos(\phi-\phi_o) - L_\phi \cos\theta \sin(\phi-\phi_o) + L_z \sin\theta]\Big\}\, e^{-i\,[\,l_1 z_o \cos\theta + l_2 \sin\theta \cos(\phi-\phi_o)]}$$

$$= \frac{l_1 l_2}{2}\sum_{n=-\infty}^{\infty} (-i)^n e^{in\phi} \sum_{p=0}^{\infty} (-i)^{p+n}$$

$$\cdot \Bigg\{ \hat{\theta}\Bigg[\left(iJ_n(l_2 \sin\theta)\frac{n\cot\theta}{l_2} K_{\phi,n} - J_n'(l_2\sin\theta)\, L_{\phi,n}^p\right) J_p(l_1\cos\theta) \tag{4-56}$$

$$- \frac{i}{2} J_n(l_2 \sin\theta)\sin\theta\, K_{z,n}^p\Big(J_p(l_1\cos\theta) + J_{p+2}(l_1\cos\theta)\Big)\Bigg]$$

$$- \hat{\phi}\Bigg[\left(J_n'(l_2\sin\theta)\, K_{\phi,n}^p + iJ_n(l_2\sin\theta)\frac{n\cot\theta}{l_2} L_{\phi,n}^p\right) J_p(l_1\cos\theta)$$

$$- \frac{i}{2} J_n(l_2\sin\theta)\sin\theta\, L_{z,n}^p\Big(J_p(l_1\cos\theta) + J_{p+2}(l_1\cos\theta)\Big)\Bigg]\Bigg\}$$

As $\theta \to 0$, only the $n = \pm 1$ terms are nonzero in this limit, and $\hat{\theta} = \hat{\rho} = \hat{x}\cos\phi + \hat{y}\sin\phi$, $\hat{\phi} = -\hat{x}\sin\phi + \hat{y}\cos\phi$. We have $\hat{\theta} \pm i\hat{\phi} = (\hat{x} \pm \hat{y})e^{\mp i\phi}$:

$$\frac{\vec{E}^{sc}(\vec{r})}{G(\vec{r})} = \frac{l_1 l_2}{4}\sum_{p=0}^{\infty} (-i)^p \Big\{\hat{x}\Big[\Big(K_{\phi,1}^p - K_{\phi,-1}^p\Big) + i\Big(L_{\phi,1}^p + L_{\phi,-1}^p\Big)\Big]$$
$$+ i\,\hat{y}\Big[\Big(K_{\phi,1}^p + K_{\phi,-1}^p\Big) + i\Big(L_{\phi,1}^p - L_{\phi,-1}^p\Big)\Big]\Big\}\, J_p(l_1) \tag{4-57}$$

Similarly, as $\theta \to \pi$, only the $n = \pm 1$ terms are nonzero. $\hat{\theta} = -\hat{\rho} = -\hat{x}\cos\phi - \hat{y}\sin\phi$, $\hat{\phi} = -\hat{x}\sin\phi + \hat{y}\cos\phi$. We have $\hat{\theta} \pm i\hat{\phi} = -(\hat{x} \mp \hat{y})e^{\pm i\phi}$:

$$\frac{\vec{E}^{sc}(\vec{r})}{G(\vec{r})} = \frac{l_1 l_2}{4}\sum_{p=0}^{\infty} i^p \Big\{\hat{x}\Big[\Big(K_{\phi,1}^p - K_{\phi,-1}^p\Big) - i\Big(L_{\phi,1}^p + L_{\phi,-1}^p\Big)\Big]$$
$$+ \hat{y}\Big[i\Big(K_{\phi,1}^p + K_{\phi,-1}^p\Big) + \Big(L_{\phi,1}^p - L_{\phi,-1}^p\Big)\Big]\Big\}\, J_p(l_1) \tag{4-58}$$

E. INSIDE AND OUTSIDE SURFACE CURRENTS

From Eq. (2-20) we have:

$$\begin{bmatrix} \vec{K}_n^{+p} - \vec{K}_n^{-p} \\ \vec{L}_n^{+p} - \vec{L}_n^{-p} \end{bmatrix} = - \begin{bmatrix} Z^{-1}\Delta & iZ^{-1}\sigma_2 \\ -i\sigma_2(Z - \Delta Z^{-1}\Delta) & -\sigma_2 \Delta Z^{-1} \sigma_2 \end{bmatrix} \begin{bmatrix} \vec{K}_n^{p} \\ \vec{L}_n^{p} \end{bmatrix} \tag{4-59}$$

Therefore the following matrix equation is obtained:

$$\begin{bmatrix} \vec{K}_n^{\pm p} \\ \vec{L}_n^{\pm p} \end{bmatrix} = \frac{1}{2} \left\{ I \mp \begin{bmatrix} Z^{-1}\Delta & iZ^{-1}\sigma_2 \\ -i\sigma_2(Z - \Delta Z^{-1}\Delta) & -\sigma_2 \Delta Z^{-1} \sigma_2 \end{bmatrix} \right\} \begin{bmatrix} \vec{K}_n^{p} \\ \vec{L}_n^{p} \end{bmatrix} \tag{4-60}$$

V. COMPUTATION AND RESULTS

The solution to the problem of the scattering of an anisotropically coated tubular cylinder of finite length as formulated in the previous chapter has been coded in FORTRAN and tested. The program listings are included in the Appendix. Computation has been carried out on the 32-bit Sun SPARC Station running under the Unix operating system in the Electrical and Computer Engineering Department and the Computer Center. The evaluation of the double series expansion coefficients of the Green's function and its derivatives for greater values of ka and kh have also been done on the 64-bit Cray Y-MP EL98 in the Visualization Lab so that the accuracy of the results can be accessed.

The program accepts the geometry of the cylinder, the surface impedance matrices, the incident wave frequency, direction and its polarization in terms of TE, TM or some linear combination of the two. It computes the sum surface currents and the far field radiation in the directions specified, including the strength and the phase of each field component. It also breaks down the sum surface currents into the outside and the inside currents.

In this chapter, some interesting results of computation for the scattering of a tubular cylinder having the length-to-diameter ratio h/a of 4 or 6 are presented. All figures are attached at the end of this chapter. The wave is incident from the positive z-axis and is polarized in the $\hat{y}$ direction.

A. COMPARISON WITH EXPERIMENTAL DATA

For backscattering from a perfectly conducting tubular cylinder of a y-polarized plane wave incident from the positive z-axis, experimental data are available [6] over a frequency band of well beyond three octaves, with the circumference-to-wavelength ratio $ka = 2\pi a/\lambda$ varied from 0.9448 to 3.3152. These data are measured with two sets of four cylinders each; one set having $h/a = 4$, the other with $h/a = 6$. Both data sets use the inner radii of the tubular cylinders as the parameter a [7]. The cylinder length-to-wavelength ratio $2h/\lambda$ varies from 1.2030 to 4.2210 for the $h/a = 4$ case and from 1.8045 to 6.3315 for the h/a

= 6 case.

The experimental data are plotted against the results of theoretical computation. Figure 5-1 shows the backscattering from the set of cylinders having *h*/*a* = 4. Figure 5-2 shows those data from the set of cylinders with *h*/*a* = 6. The output of theoretical computation follows the measured data very closely. Note that the cutoff frequency of the dominant circular waveguide mode, TE_{11}, occurs at $2h/\lambda = 2.344$ for the cylinders with *h*/*a* = 4 and at $2h/\lambda = 3.516$ for those with *h*/*a* = 6.

B. NULL ON-AXIS BACKSCATTERING

Three different ways of coating a surface impedance Z_s on a tubular cylinder of *h*/*a* = 6 are considered: Both on the outside surface and the inside surface; Only on the inside and leaving the outside surface perfectly conducting; Only on the outside and leaving the inside surface perfectly conducting. Here Z_s has the elements $z_{s11} = 0.5$, z_{s22} varies from 0.1 to 5, $z_{s12} = z_{s21} = 0$. At a fixed frequency for which $2h/\lambda = 3.194$, slightly below the TE_{11} circular waveguide dominant mode cutoff of 3.516, the scattered fields are plotted in Figures 5-3 through 5-5. Figure 5-3 shows the results of computation for the case $Z^+ = Z^- = Z_s = Z$. As $_{22}$ is varied through 2, the backscattering cross section vanishes as predicted in Chapter 3. Figure 5-4 shows the results for the case $Z^+ = 0$ and $Z^- = Z_s$. As the impedance on the inside surface is increased, the excited field inside the tubular cylinder is dissipated and the backscattered power decreases exponentially. Figure 5-5 shows the results for the case $Z^+ = Z_s$ and $Z^- = 0$. The backscattered power drops off rapidly at first as the impedance on the outside surface is increased. But the cross section quickly settles down to a fixed value presumably due to the current excited on the perfectly conducting inside surface of the cylinder.

Results of computation for the same configurations but at a higher frequency for which $2h/\lambda = 4.865$, above the TE_{11} circular waveguide dominant mode cutoff of 3.516, are plotted in Figures 5-6 through 5-8. Now that the incident wave can propagate through the cylinder in the dominant waveguide mode, the backscattering cross sections are about an order of magnitude smaller than in previous cases. Figure 5-6 shows the results of computation for the case $Z^+ = Z^- = Z_s = Z$. Again the backscattering cross section vanishes as z_{22} is varied through 2. Figure 5-7 shows the results for the case $Z^+ = 0$ and $Z^- = Z_s$ while Figure 5-8 shows the results for the case $Z^+ = Z_s$ and $Z^- = 0$. It appears that, above cutoff, the contribution to the backscattering cross section from the inside of the tubular cylinder is minimal: once the outside current is reduced by the increase in surface impedance, the backscattering cross section is reduce by more than 10 dB as shown in Figure 5-8. When the impedance coating is applied in the inside surface, the maximal reduction in the backscattering cross section is only about 1.2 dB.

C. FREQUENCY DEPENDENCE

The axial backscattering of the two cases when the tubular cylinder is coated only on the inside or only on the outside with $z_{s11} = 0.5$ and $z_{22} = 2$ are investigated for different frequencies with $2h/\lambda$ varying from 0.1 to 7.5. Figure 5-9 shows the results of the case when only the inside surface is coated so that the backscattering is mainly due to the current excited on the outside surface. The reflection from the ends of the cylinder causes the fluctuation in backscattering cross section. Being waves in free space on the outside of the cylinder, the maxima and minima are evenly spaced with the minima occurring when $2h/\lambda$ is a multiple of half integer. Figure 5-10 shows the results when only the outside surface is coated and the current on the inside surface dominates the contribution. The distinct feature in this case is that the backscattering cross section does not fluctuate with varying frequency below the waveguide mode cutoff. The incident wave is able to penetrate deeper into the cylinder with increasing frequency, resulting in a constantly rising strength of the backscattered field. Once beyond the circular waveguide mode cutoff, the wave can pass through the cylinder in the TE_{11} mode and the backscattering diminishes. The oscillation in the cross section at these higher frequencies represents the interference of reflected waves at the ends of the tube and the separation between maxima and minima should be determined by the guide wavelength at the particular frequency. These two situations should be compared to Figure 5-11 which shows the results when both sides of the cylinder are perfectly conducting and the current can flow freely. The distinct notch in the cross section near the TE_{11} mode cutoff at $2h/\lambda = 3.516$ and the subsequent faster variation in the cross section shows the combination of the two distinct features of Figures 5-9 and 5-10.

D. COMPUTATION ACCURACY

The main difficulty encountered in the computation is the evaluation of $G_n^{p,q}(l_1, l_2)$ and its l_2-derivative by double power series sum when l_1 becomes large. Computations for Figure 5-11, 5-13 and 5-15 use the $G_n^{p,q}$ values evaluated with the Cray computer which has a 128-bit double precision number. Computations for Figures 5-12, 5-14 and 5-16 use the $G_n^{p,q}$ values evaluated with the Sun SPARC Station which has a 64-bit double precision number. For $2h/\lambda$ greater than about 6.2, the SPARC Station fails to provide accurate results.

Figures 5-13 and 5-15 are axial backscattering from a cylinder of $h/a = 6$ coated with the impedances having the elements $z_{11}^+ = z_{22}^- = 0.5$, $z_{22}^+ = z_{11}^- = 0.4$, $z_{12}^+ = z_{21}^+ = z_{12}^- = z_{21}^- = 0.3$. Figure 5-13 shows the co-polarized backscattered field while Figure 5-15 shows the cross-polarized backscattering.

Cylinder	$2h(cm)$	$2a^{+}$	$2a^{-}$
1	3.566	0.9525	0.8915
2	4.796	1.27	1.199
3	6.064	1.588	1.516
4	7.396	1.908	1.849

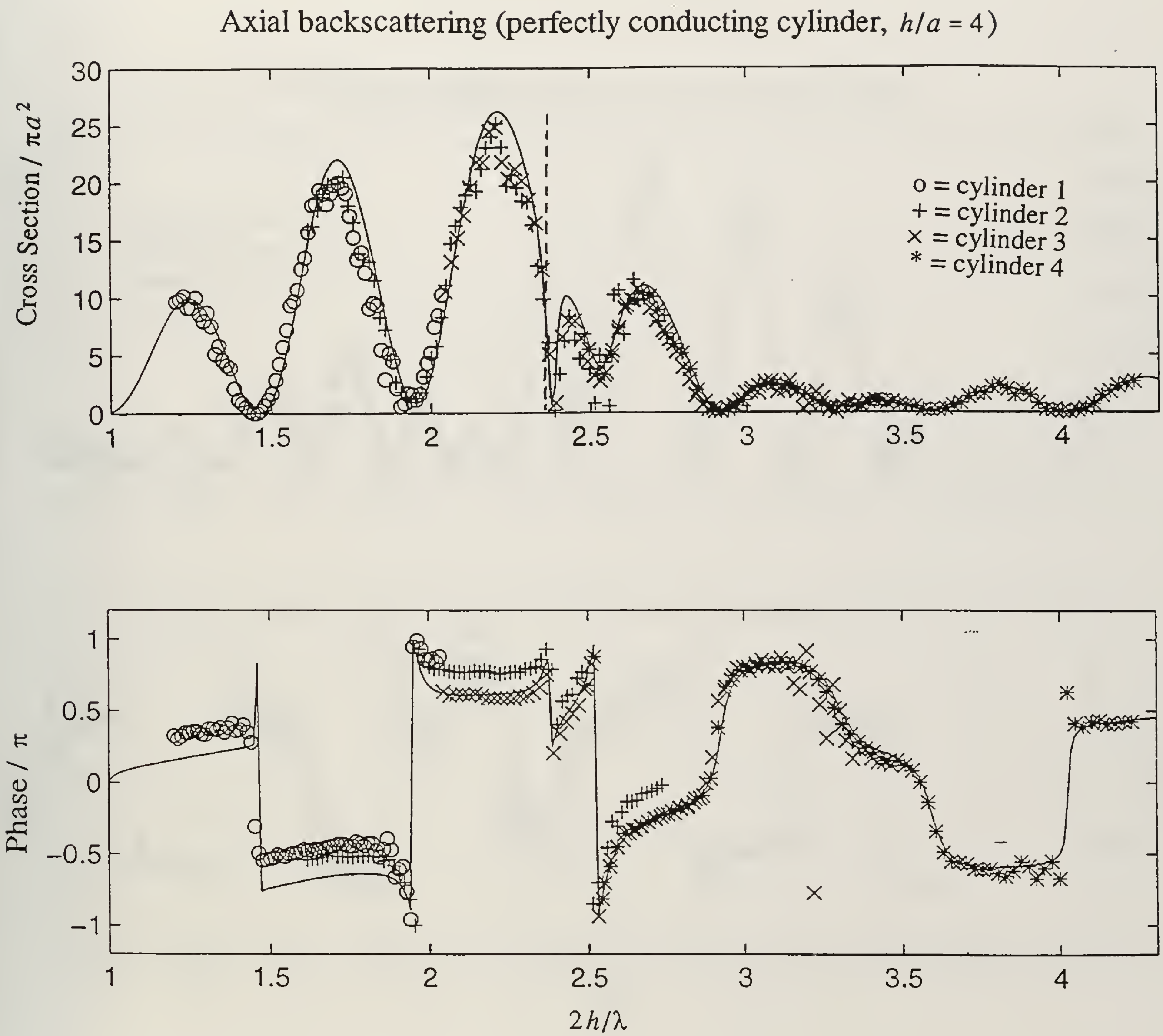

Figure 5-1.

Cylinder	*2h(cm)*	$2a^{+}$	$2a^{-}$
1	5.394	0.9525	0.8915
2	7.193	1.270	1.199
3	9.098	1.588	1.516
4	11.09	1.905	1.849

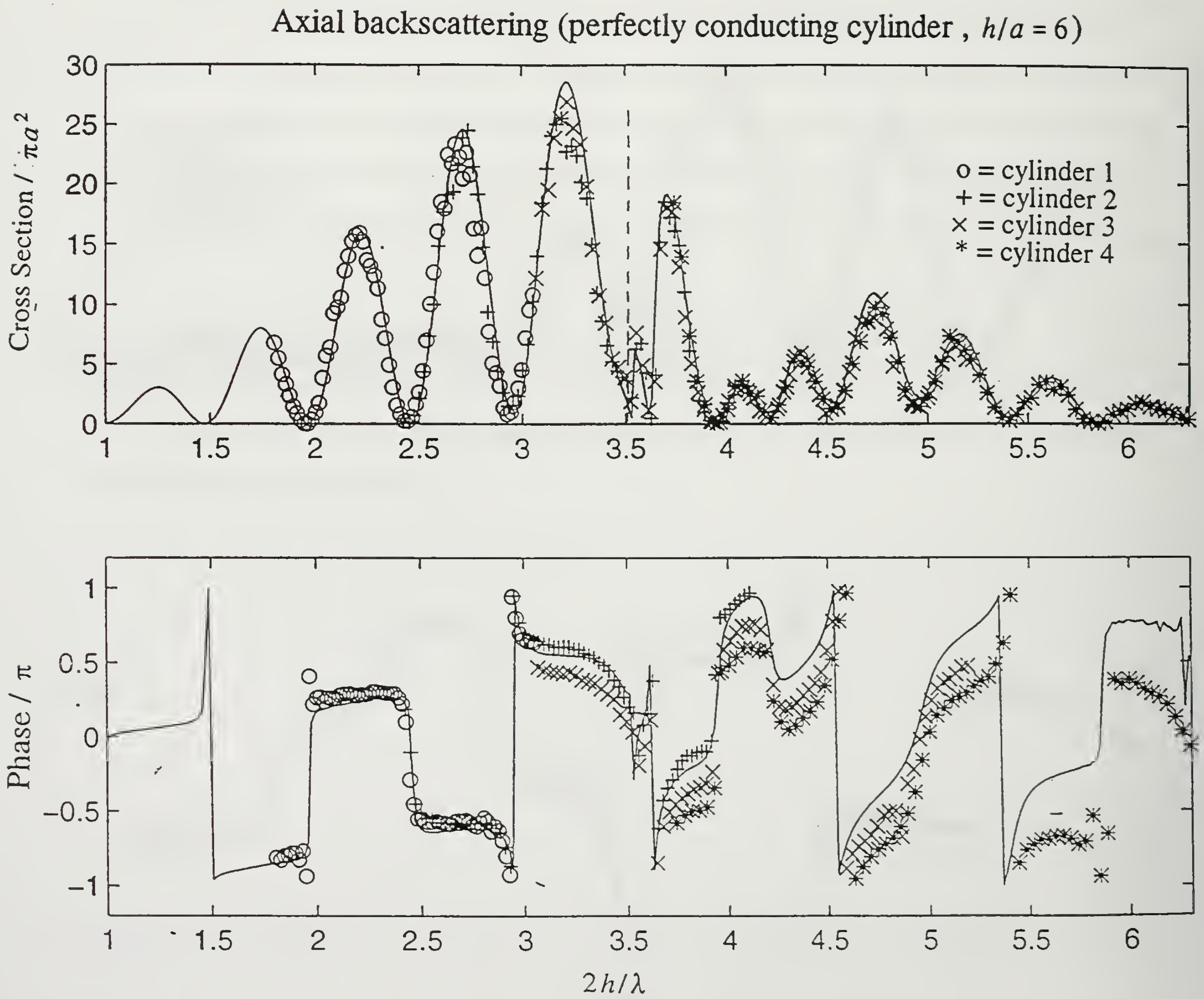

Figure 5-2.

$$Z^{+} = Z^{-} = Z_s = \begin{bmatrix} 0.5 & 0 \\ 0 & z_{s22} \end{bmatrix}$$

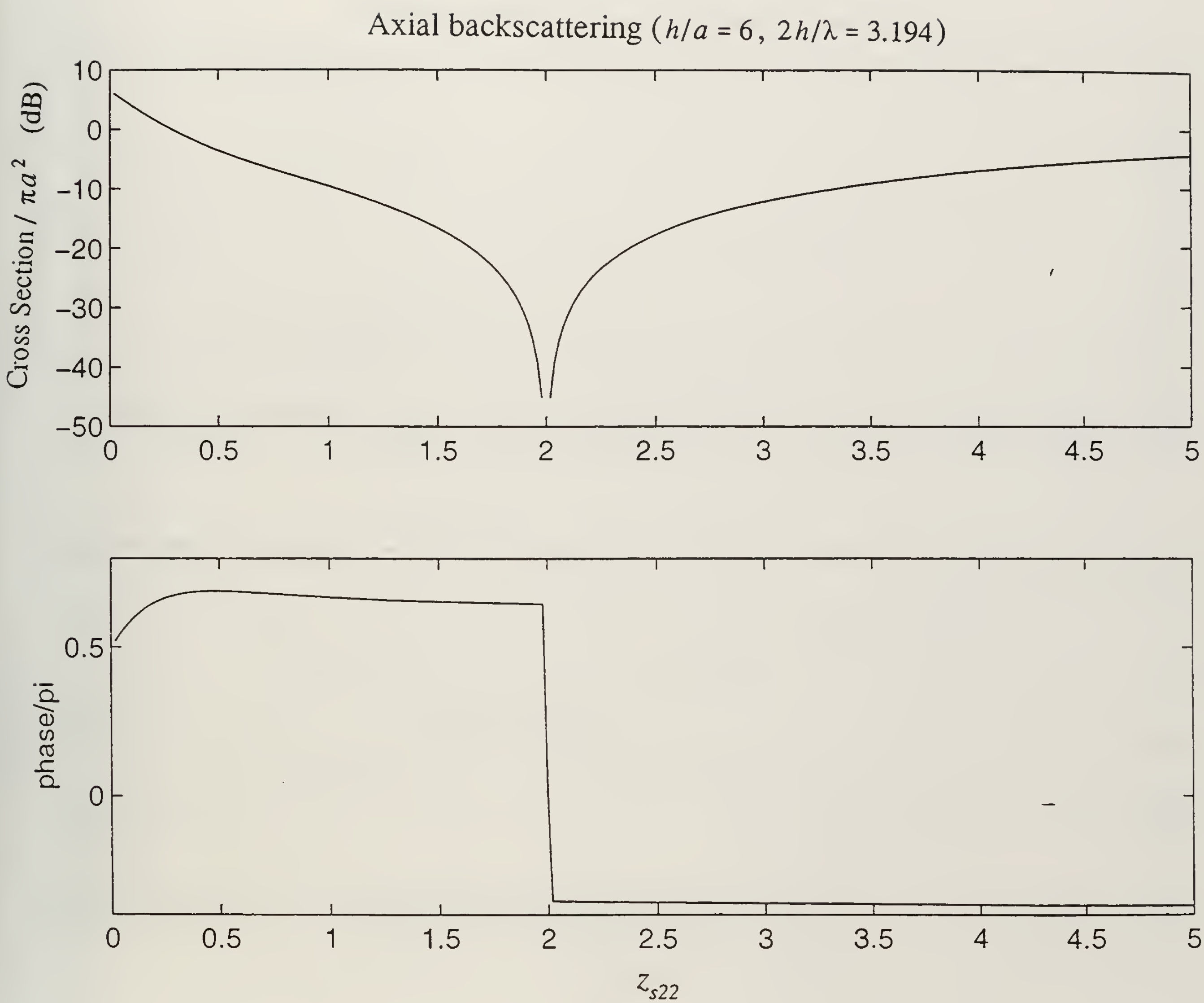

Figure 5-3.

$$Z^{+} = 0, \quad Z^{-} = Z_s = \begin{bmatrix} 0.5 & 0 \\ 0 & z_{s22} \end{bmatrix}$$

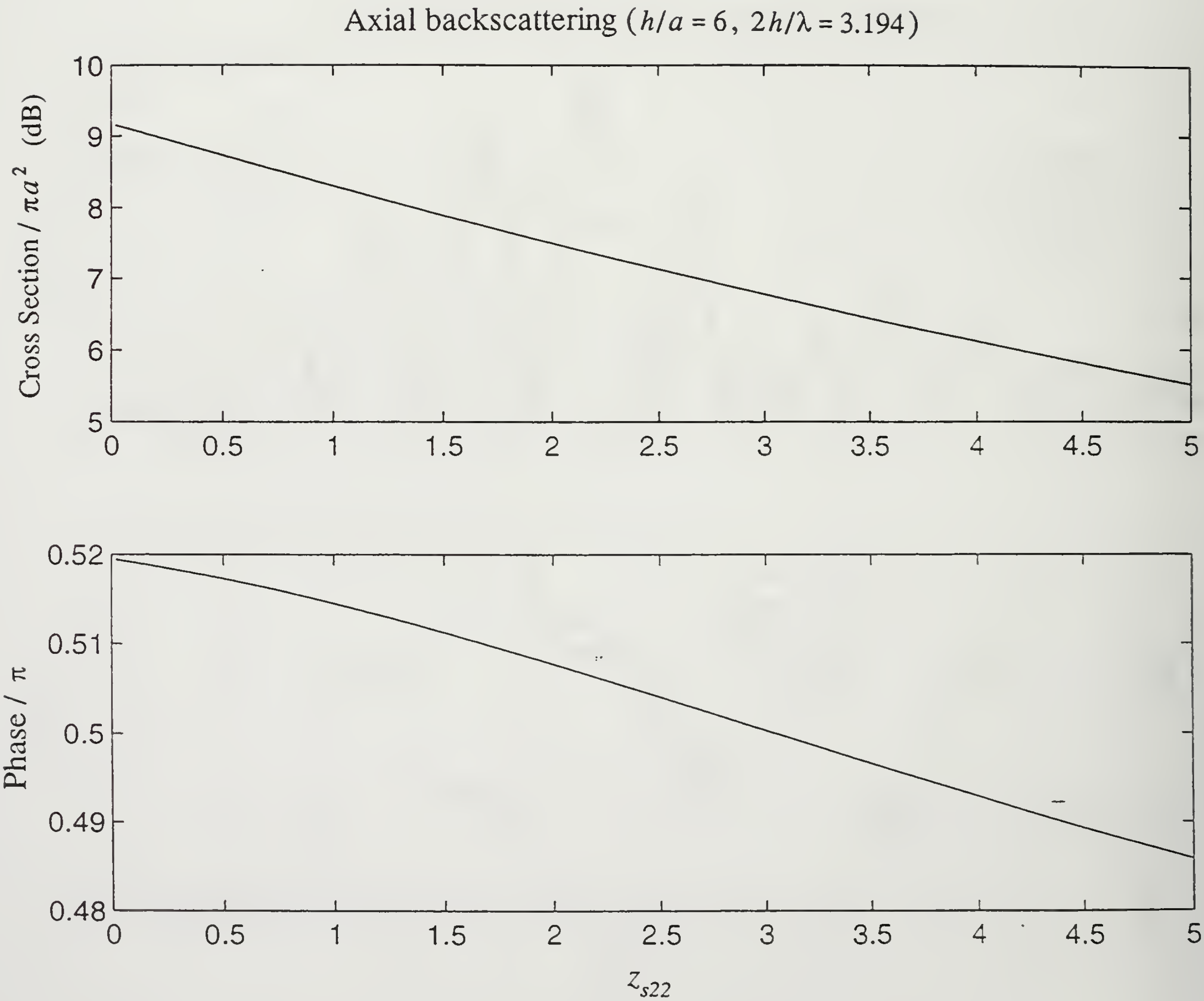

Figure 5-4.

$$Z^{-} = 0, \quad Z^{+} = Z_s = \begin{bmatrix} 0.5 & 0 \\ 0 & z_{s22} \end{bmatrix}$$

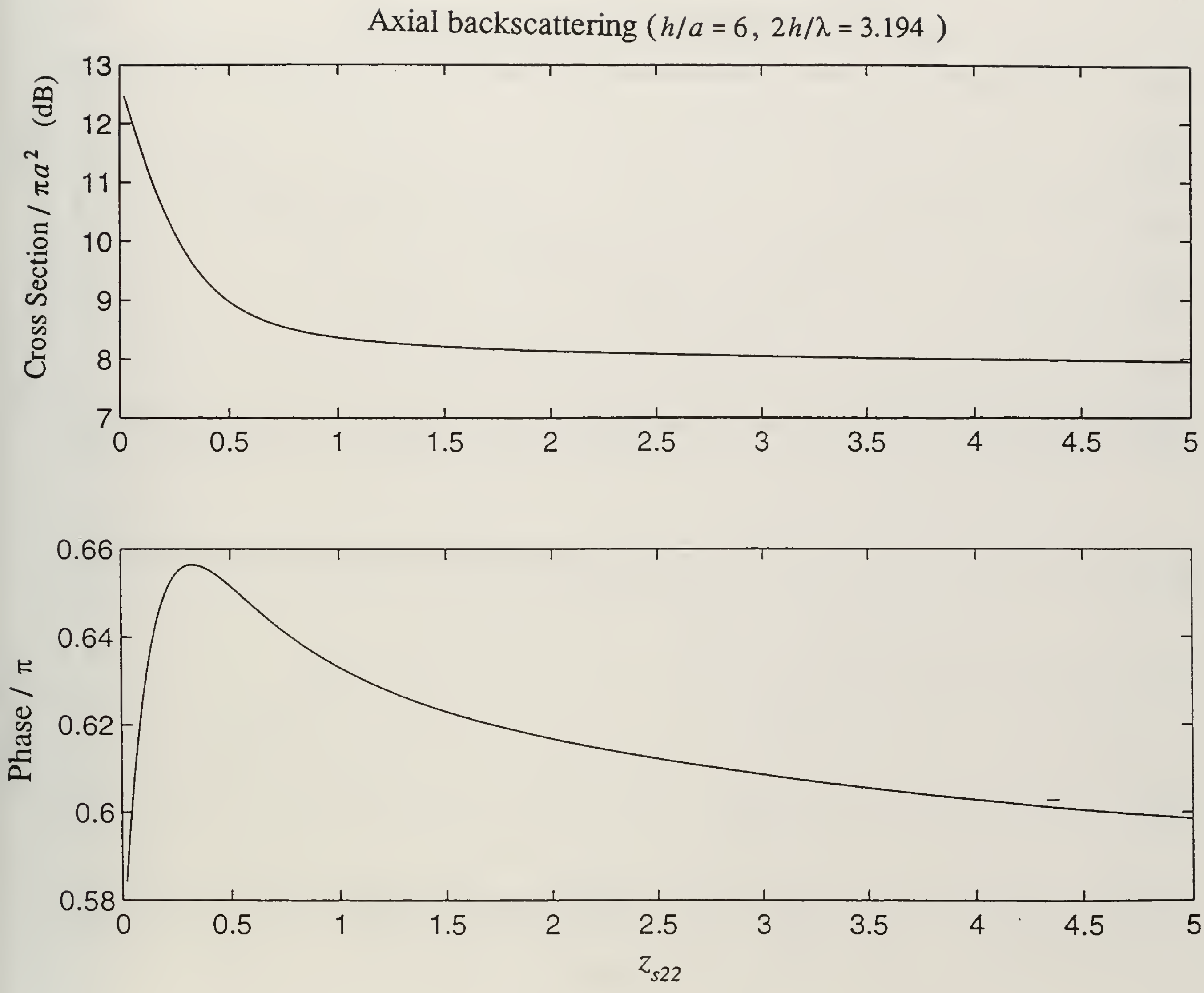

Figure 5-5.

$$Z^{+} = Z^{-} = Z_s = \begin{bmatrix} 0.5 & 0 \\ 0 & z_{22} \end{bmatrix}$$

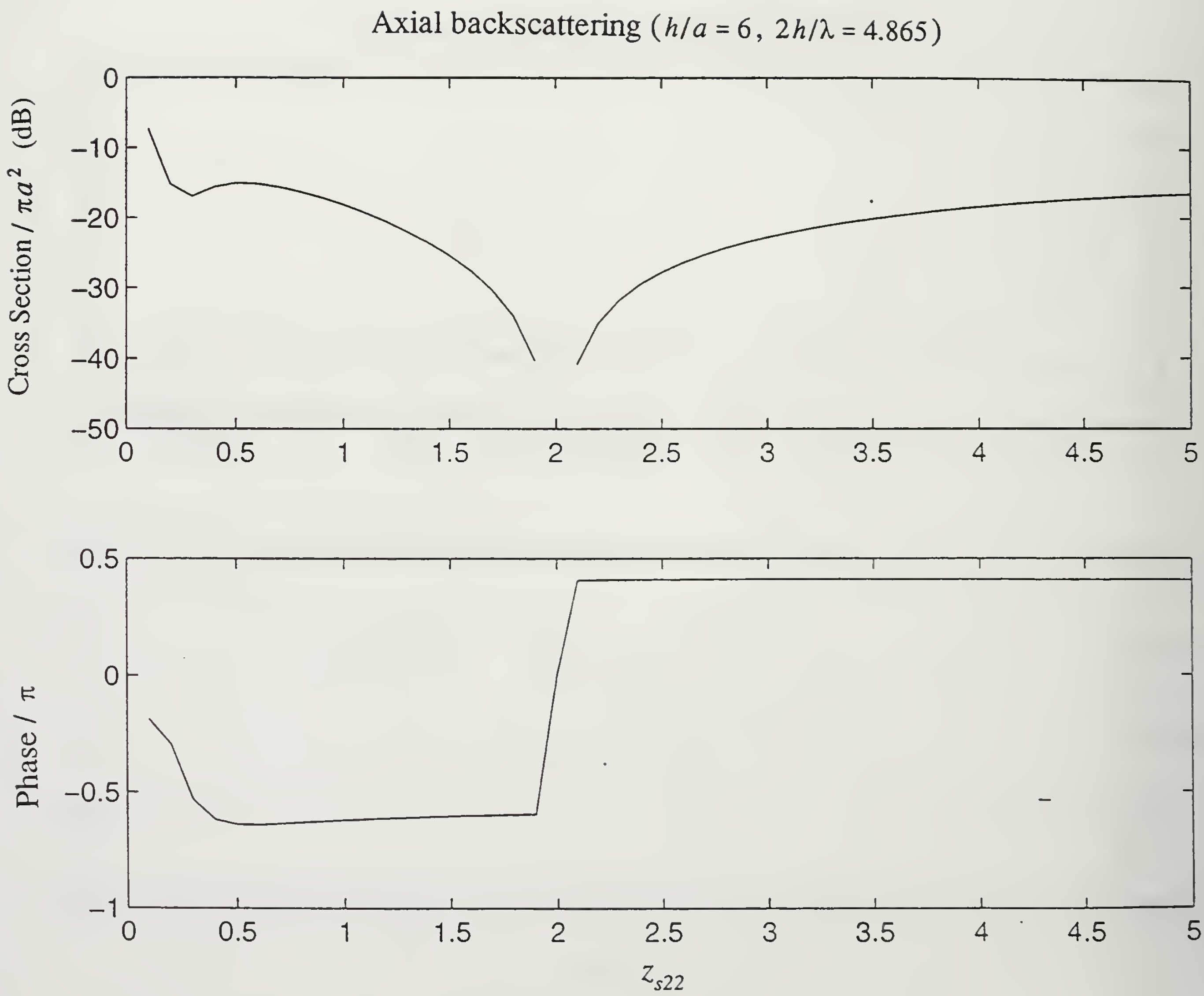

Figure 5-6.

$$Z^{+} = 0\,, \quad Z^{-} = Z_s = \begin{bmatrix} 0.5 & 0 \\ 0 & z_{s22} \end{bmatrix}$$

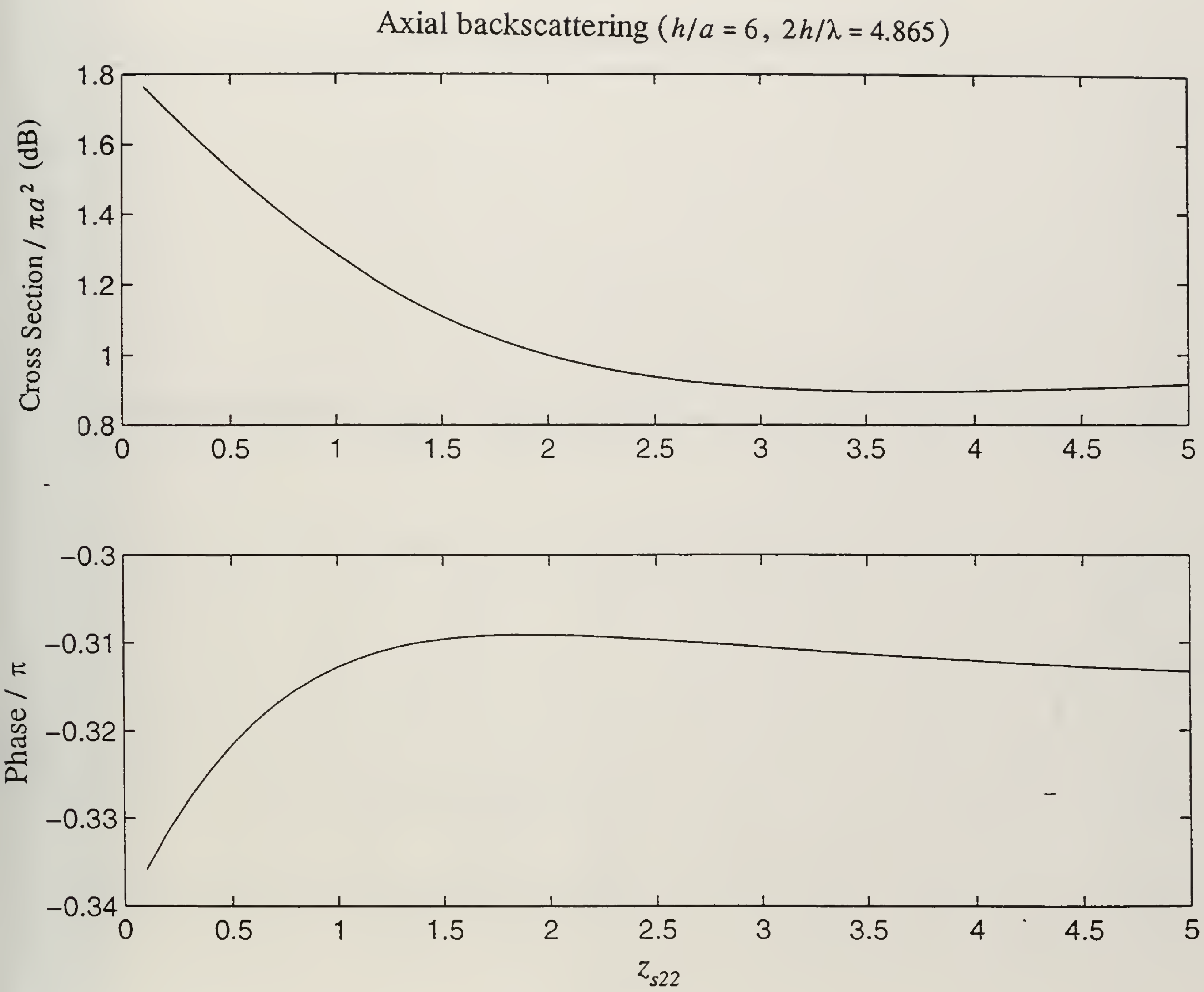

Figure 5-7.

$$Z^- = 0, \quad Z^+ = Z_s = \begin{bmatrix} 0.5 & 0 \\ 0 & z_{s22} \end{bmatrix}$$

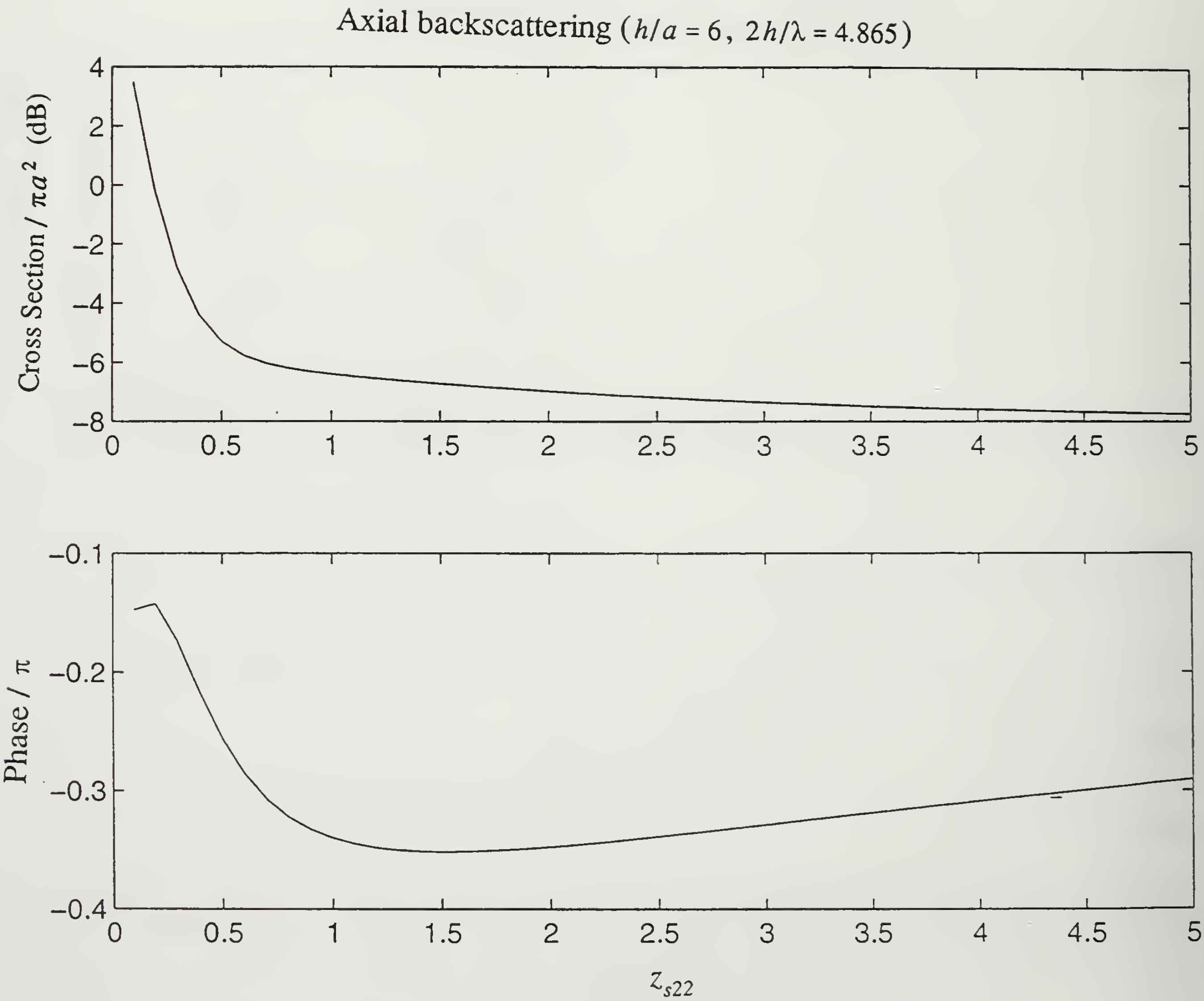

Figure 5-8.

$$Z^{+} = 0, \quad Z^{-} = Z_s = \begin{bmatrix} 0.5 & 0 \\ 0 & 2.0 \end{bmatrix}$$

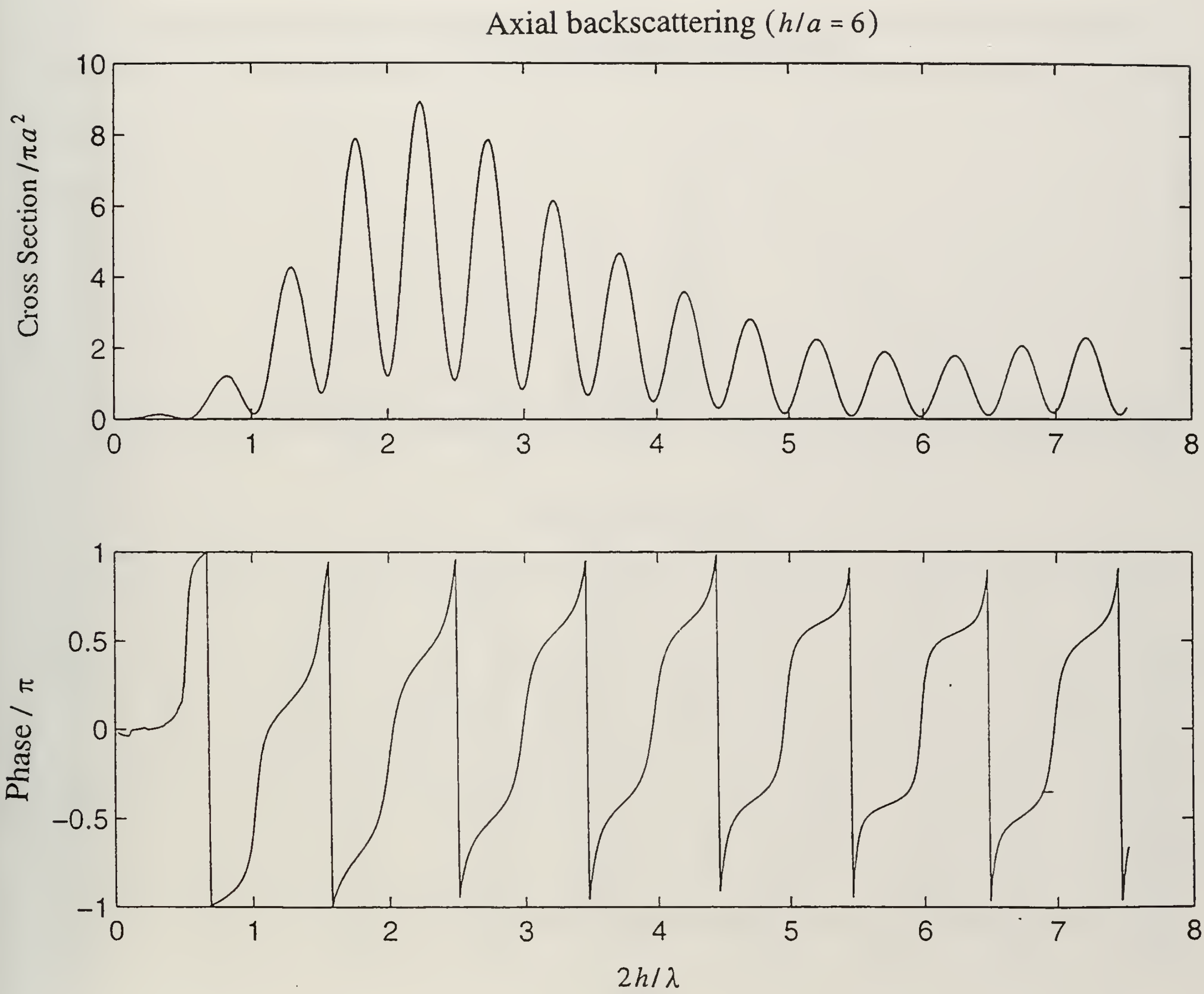

Figure 5-9.

$$Z^- = 0\,, \quad Z^+ = Z_s = \begin{bmatrix} 0.5 & 0 \\ 0 & 2.0 \end{bmatrix}$$

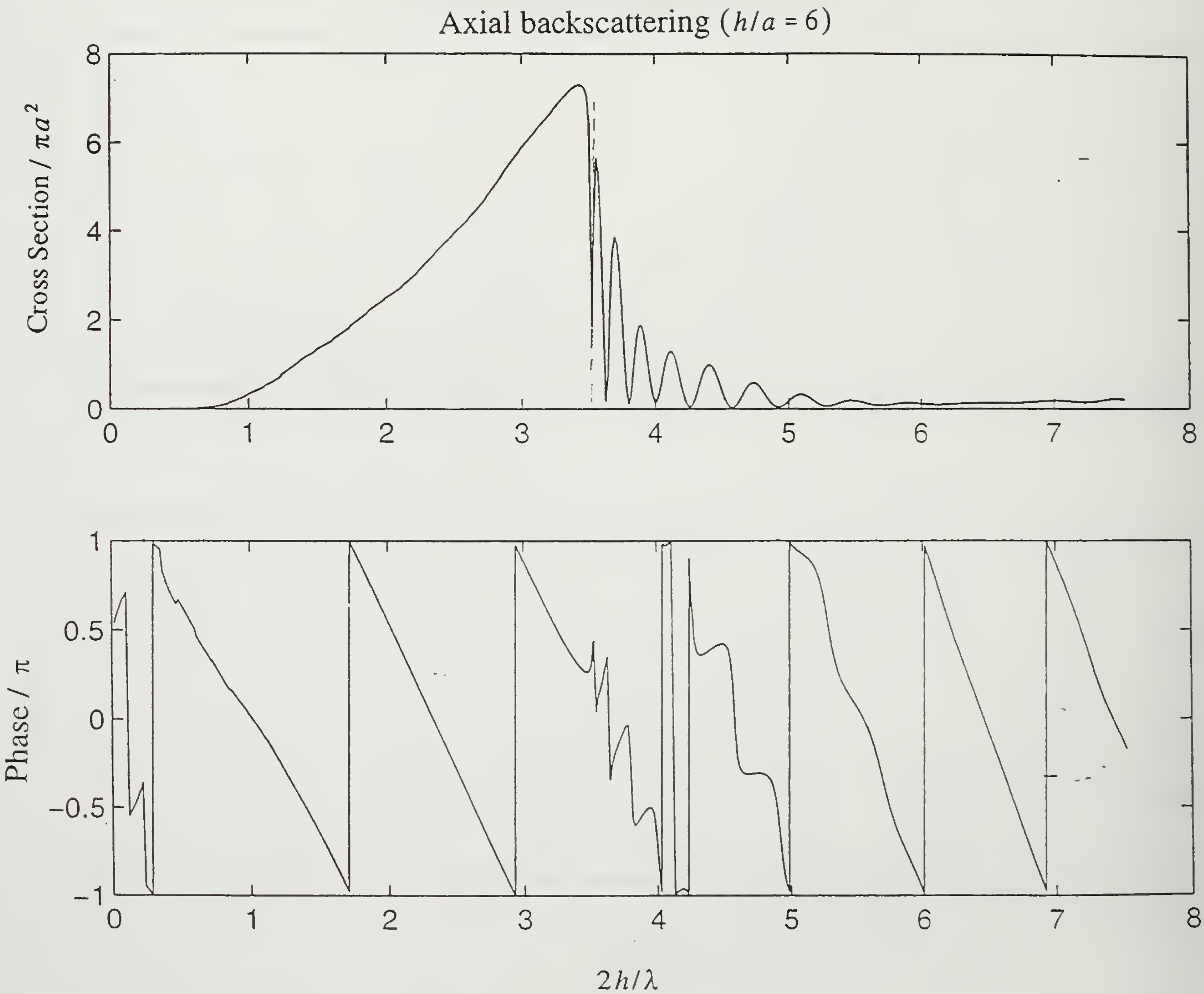

Figure 5-10.

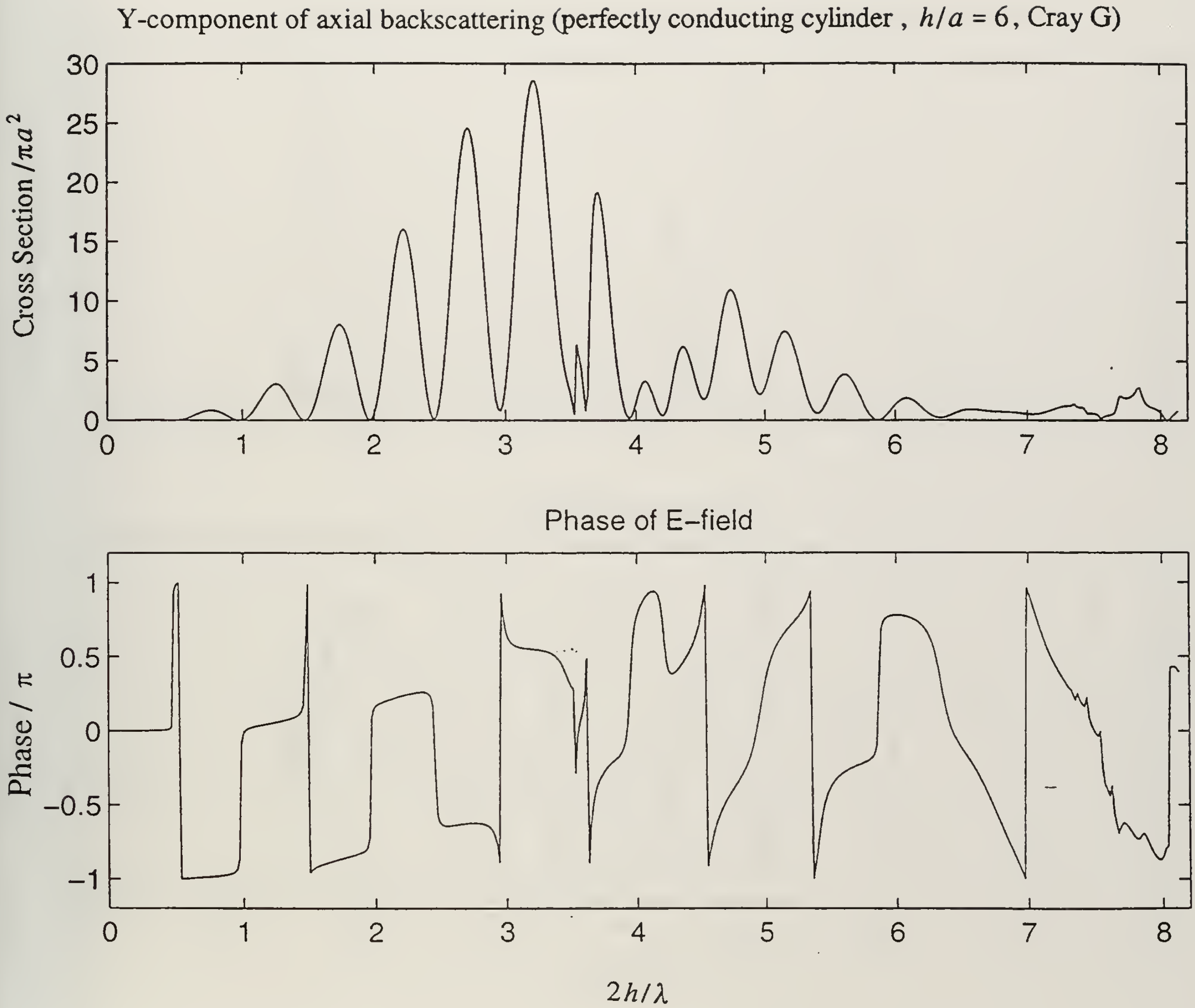
Y-component of axial backscattering (perfectly conducting cylinder , $h/a = 6$, Cray G)
Cross Section $/\pi a^2$
30
25
20
15
10
5
0
0
1
2
3
4
5
6
7
8
Phase of E–field
Phase / π
1
0.5
0
−0.5
−1
0
1
2
3
4
5
6
7
8
$2h/\lambda$

Figure 5-11

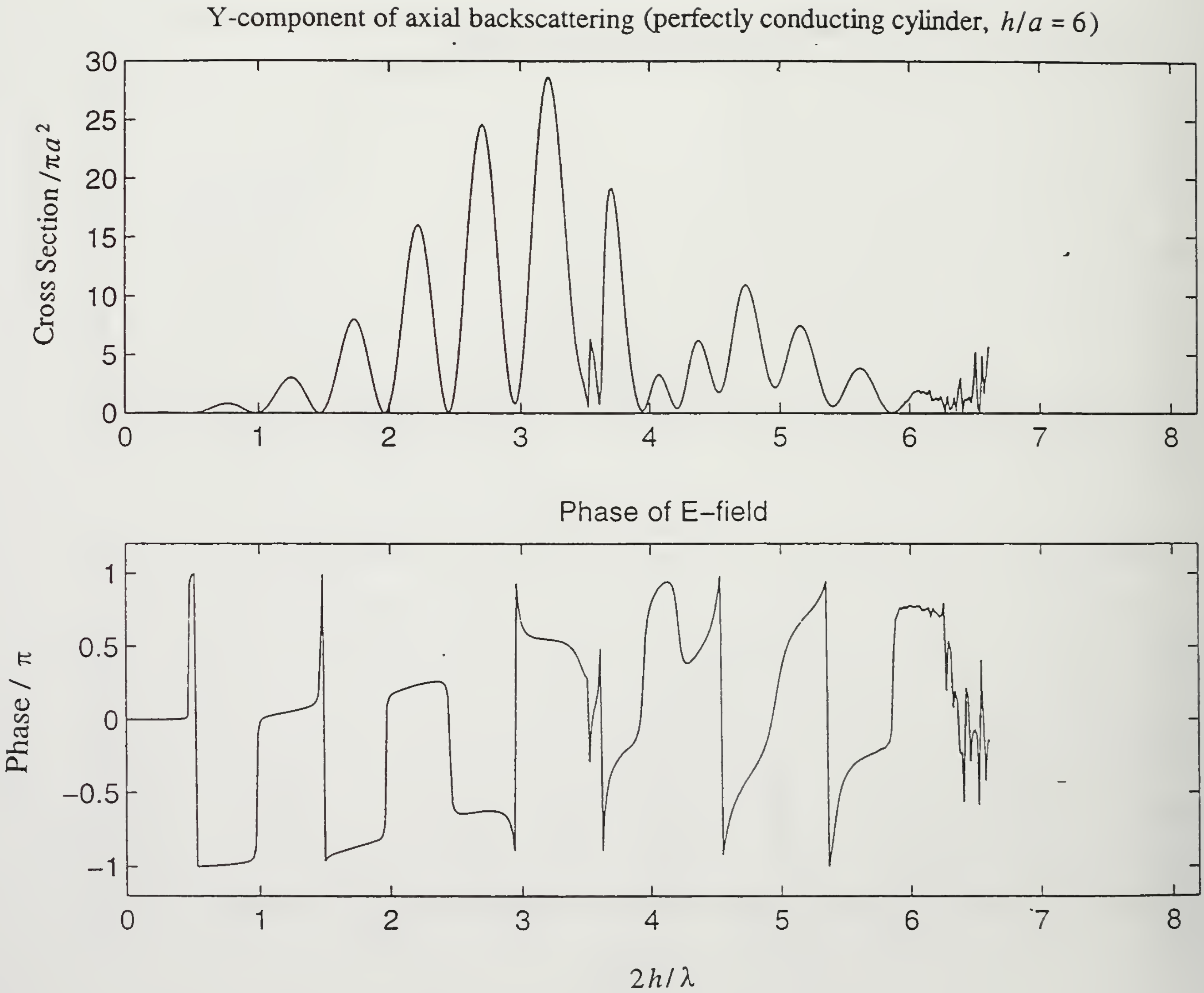
Y-component of axial backscattering (perfectly conducting cylinder, $h/a = 6$)
Cross Section $/\pi a^2$
30
25
20
15
10
5
0
0 1 2 3 4 5 6 7 8
Phase of E–field
Phase / π
1
0.5
0
−0.5
−1
0 1 2 3 4 5 6 7 8
$2h/\lambda$

Figure 5-12.

$$Z^{+} = \begin{bmatrix} 0.5 & 0.3 \\ 0.3 & 0.4 \end{bmatrix} \quad Z^{-} = \begin{bmatrix} 0.4 & 0.3 \\ 0.3 & 0.5 \end{bmatrix}$$

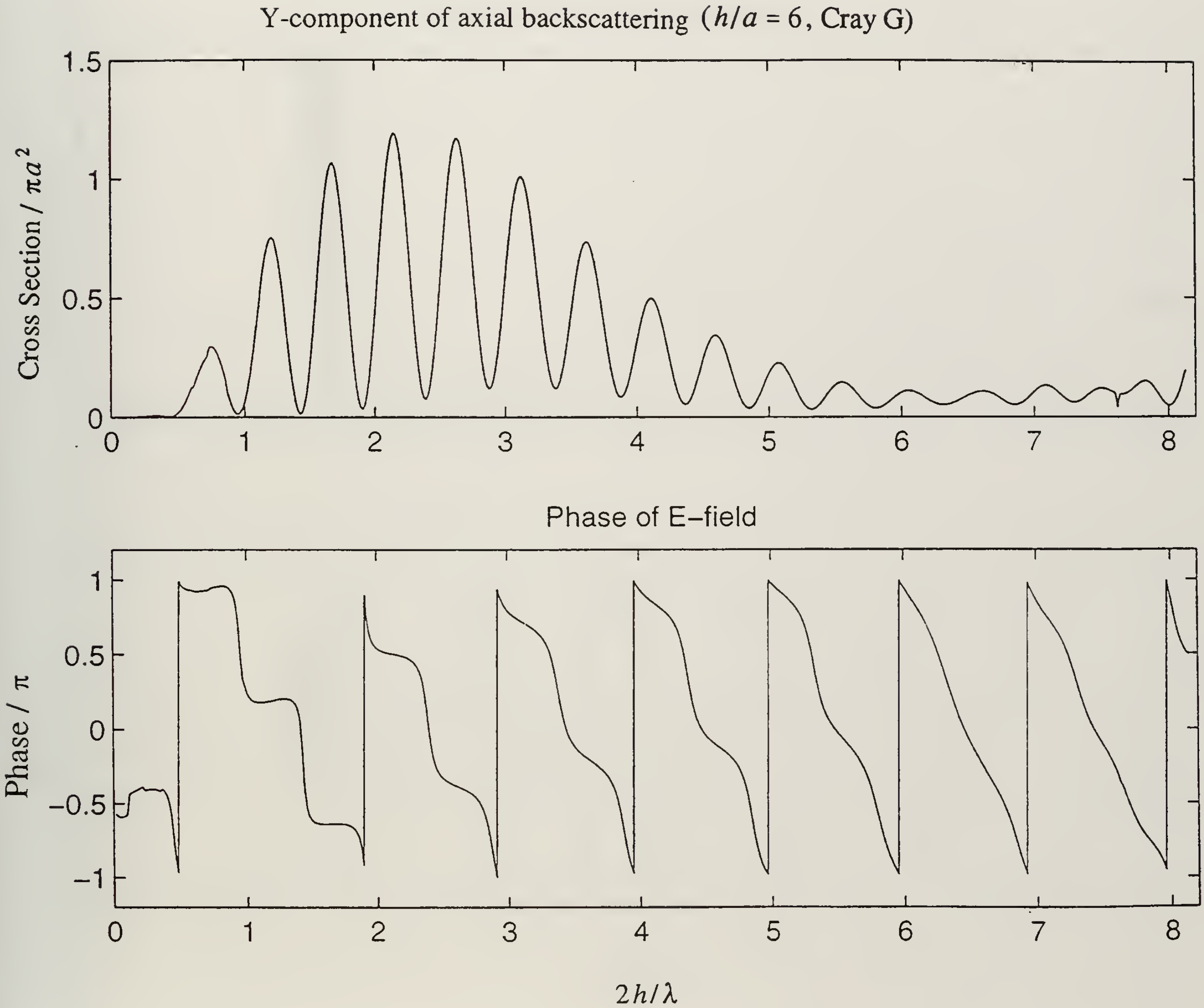

Figure 5-13.

$$Z^{+} = \begin{bmatrix} 0.5 & 0.3 \\ 0.3 & 0.4 \end{bmatrix} \quad Z^{-} = \begin{bmatrix} 0.4 & 0.3 \\ 0.3 & 0.5 \end{bmatrix}$$

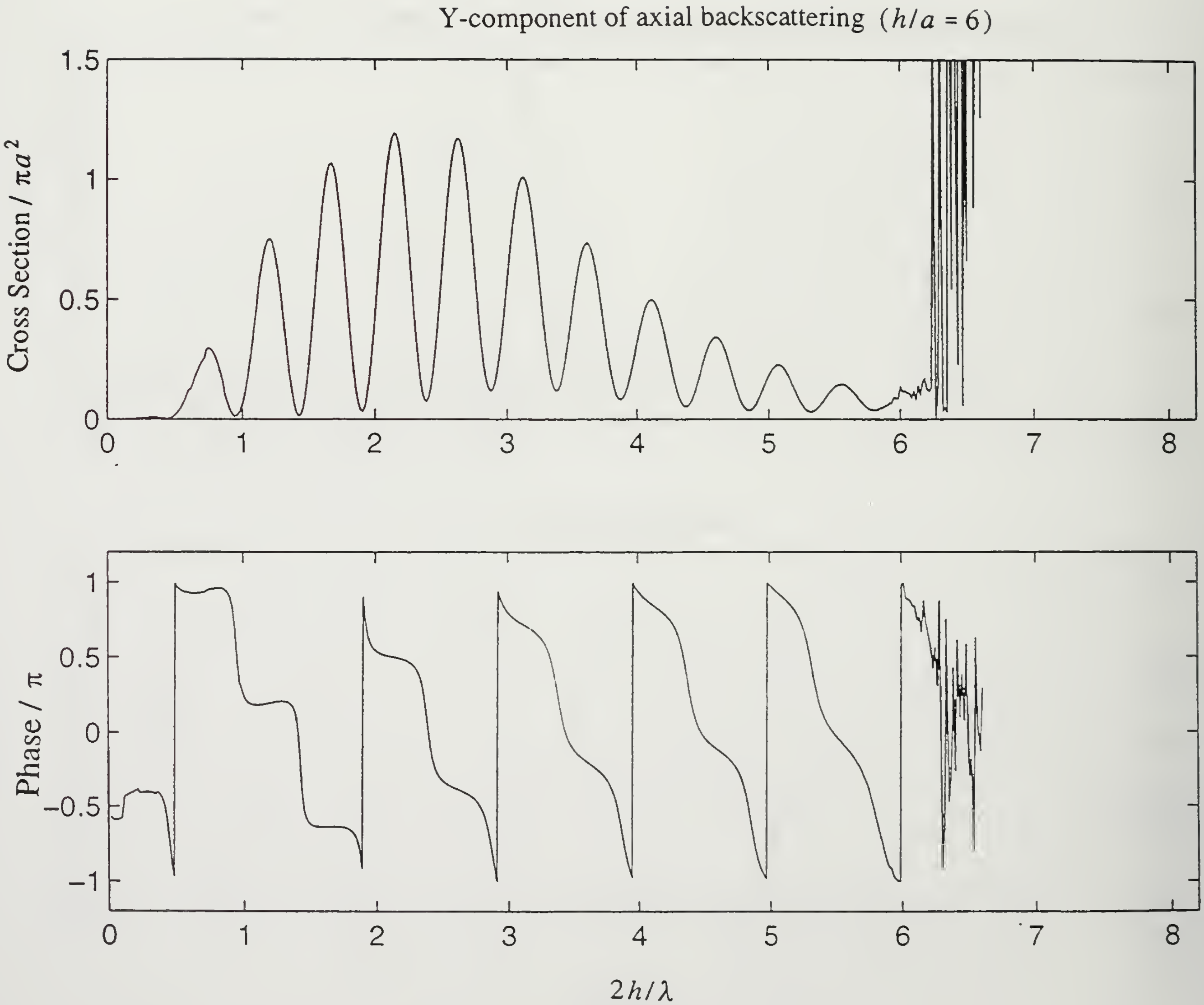

Figure 5-14.

$$Z^{+} = \begin{bmatrix} 0.5 & 0.3 \\ 0.3 & 0.4 \end{bmatrix} \quad Z^{-} = \begin{bmatrix} 0.4 & 0.3 \\ 0.3 & 0.5 \end{bmatrix}$$

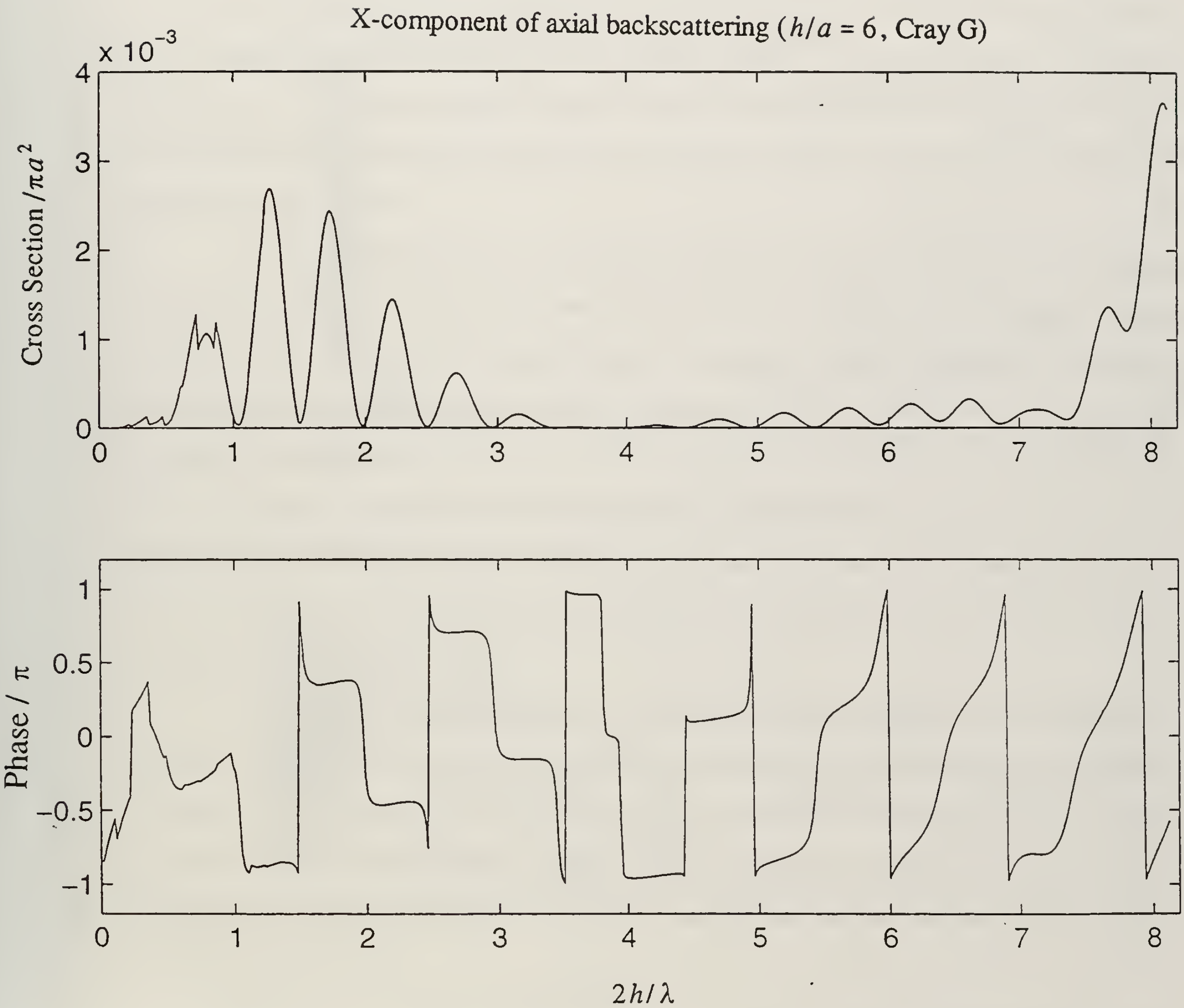

Figure 5-15.

$$Z^{+} = \begin{bmatrix} 0.5 & 0.3 \\ 0.3 & 0.4 \end{bmatrix} \quad Z^{-} = \begin{bmatrix} 0.4 & 0.3 \\ 0.3 & 0.5 \end{bmatrix}$$

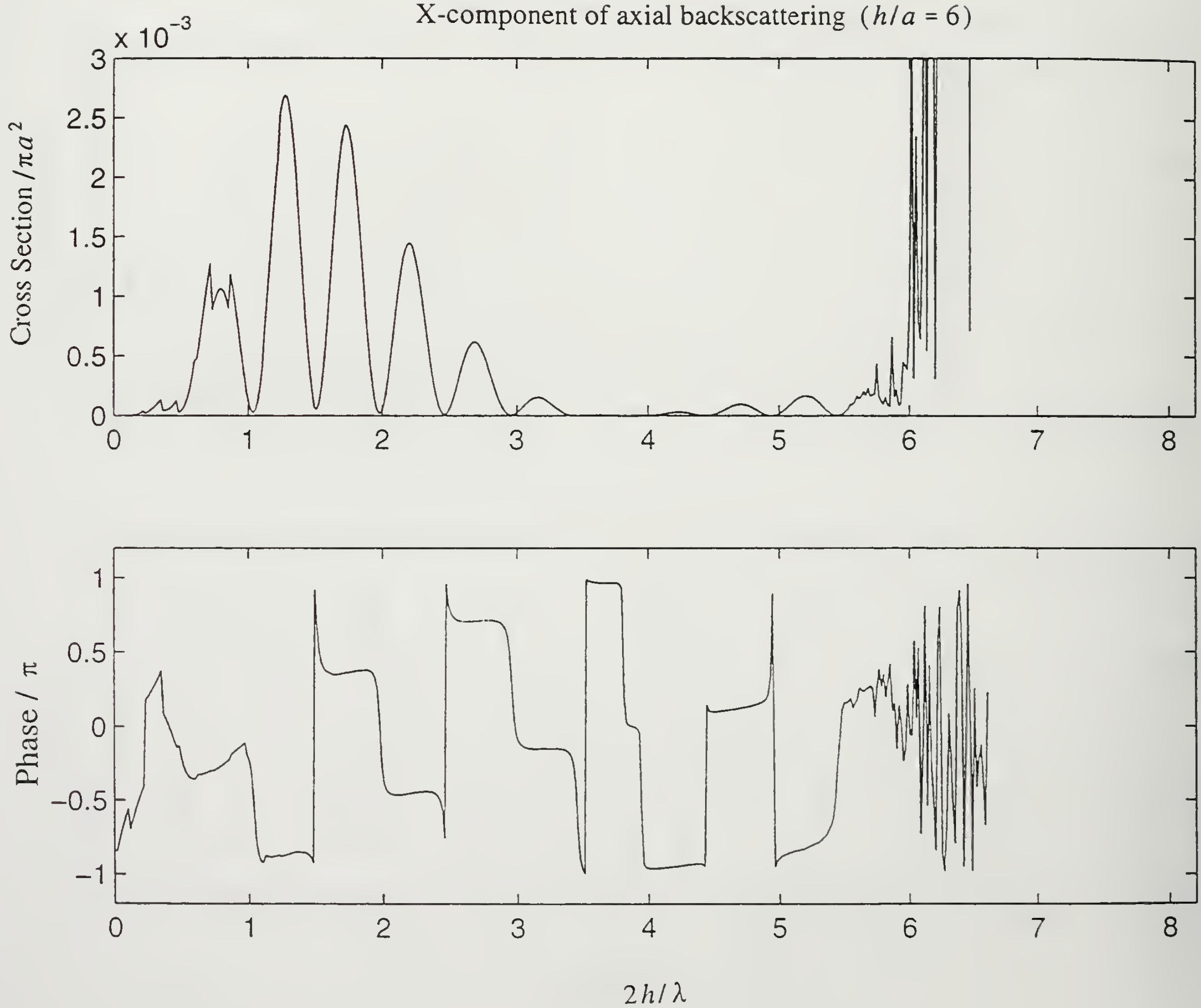

Figure 5-16.

VI. CONCLUSIONS

In this thesis, the sum-difference surface current formulation is introduced for solving electromagnetic boundary value problems when impedances are specified on both sides of a surface. For an impedance coated body, the body can be treated as being a surface separating the space into two regions of identical medium. For an exterior problem, the impedance normalized to the medium on the inside surface, Z^-, can be chosen arbitrarily; and for an interior problem, that on the outside surface, Z^+, can be arbitrary. The choice when $Z^- = -Z^+$ is of particular interest because the integrodifferential equation has only the sum of the equivalent electric surface currents on the outside and the inside surfaces as its unknown to be solved.

This formulation preserves the duality nature of Maxwell's equations and carries it over into the algebraic form of the integrodifferential operators in the equations for the sum currents. Since a 90° rotation is equivalent to undergoing a duality transform for an incident plane wave, this particular symmetry in the algebraic form of the operators leads to the sufficient conditions that if $Z^+ = Z^- = \pm\sigma_2$, or if Z^+ and Z^- are symmetric and $\det Z^+ = \det Z^- = 1$, the on-axis backscattering of an anisotropic impedance coated scatterer having a 90° rotational symmetry will be eliminated. Note that in the symmetric case, Z^+ and Z^- may vary with location. This is an extension of Weston's result [4] for which the surface impedance is isotropic.

A FORTRAN program has been written which accepts the geometry of the cylinder, the surface impedance matrices, the incident wave frequency, direction and its polarization in terms of TE, TM or some linear combination of the two. It computes the sum surface currents and the far field radiation in the directions specified, including the strength and the phase of each field component. It also breaks down the sum surface currents into the outside and the inside currents.

The results of computation using this program agree with measured data of backscattering from conducting tubular cylinders over a frequency band of more than three octaves. For a cylinder coated with surface impedance matrices satisfying the criteria for null on-axis backscattering, the numerical computation also validated the theoretical assertion.

Difficulties have been encountered about the computational accuracy in the evaluation of the double series Chebyshev expansion coefficients of the Green's function $G_n^{p,q}(l_1, l_2)$ and its l_2-derivative by double power series sum when the length of the cylinder is large compared to the wavelength: a compiler with a 64-bit double precision number can only handle a cylinder having a length up to about 6.2 wavelengths. Further work to explore the feasibility of asymptotic evaluation of these coefficients is recommended.

APPENDIX PROGRAM LISTING

A. INCLUDE FILES

~~~~~~

REALTP.INC

```
C    TYPE STATEMENTS FOR REAL AND INTEGERS AND DEFINITIONS OF CONSTANTS.
      IMPLICIT DOUBLE PRECISION (A, B, D-H, O-Z)
      IMPLICIT INTEGER*4 (I-N)
      PARAMETER (PI=3.14159265358979323846264338327D0,PI2=PI+PI,
     +           PISQ=PI*PI)
      PARAMETER (ONE=1.D0,TWO=2.D0,THR=3.D0,HXD=16.D0,ZERO=0.D0,
     +           DEGPI=180.D0,EPS8=2.220446049250313D-16)
      PARAMETER (ONEN=-ONE,HALF=ONE/TWO,THIR=ONE/THR,QUAR=HALF*HALF)
```

~~~~~~

CMPXPT.INC

```
C    IMPLICIT TYPE STATEMENT AND CONSTANTS FOR COMPLEX NUMBERS.
      IMPLICIT DOUBLE COMPLEX (C)
      PARAMETER (CZERO=(0.d0,0.d0),CONE=(1.d0,0.d0))
      PARAMETER (CONEN=(-1.d0,0.d0),CI1=(0.d0,1.d0),CI2=(0.d0,-1.d0))
```

~~~~~~

B. INPUT DATA FILES

~~~~~~

CLYGEOM.PRM

```
      30          Maximum kh (integer)
      5           Maximum ka (integer)
      8           NREGNS (integer)
```

~~~~~~

CYLFLPT.PRM

```
      1024      floating point zero I0BIT (1024 bits)
      64        floating point precision IFPBIT (64 bits)
```

~~~~~~

INPUTDAT.PRM

```
      .1D-1       DKA0, minimum ka (REAL) in the computation
      .875D-2     DINCA, increment of ka (REAL)
       40         NSTR, start point (INTEGER) in the computation
       479        NEND, end point (INTEGER) in the computation
       6.D0       RHA, ratio of h and a- (REAL)
       1          IE, if incident wave is TE-polarized it is 1, otherwise 0
       0          IM, if incident wave is TM-polarized it is 1, otherwise 0
       0.D0       THETAI, incident angle (REAL) is limited to 0 to 90 degree
       6           NTHTAO, total number (INTEGER) of angle of THATA
       3          NPHI, total number (INTEGER) of angle of PHI0 to be computed
       0.D0       THETAO, initial theta angle (REAL)
        1.D0       DELTHO, increment of theta angle (REAL)
        0.DO       PHIA, initial theta angle (REAL)
        12.D0      DELPHI, increment of phi angle (REAL)
       0          IK, set 1 to compute scattering currents on outer and inner surface
```

~~~~~~

IMPEDNCE.PRM

```
        0                IZ, if perfect conducting, IZ=0; otherwise, IZ=1.
        (.5D0,0.D0)      Impedance Z+ (phi phi)
        (.4D0,0.D0)      Impedance Z- (phi phi)
        (.3D0,0.D0)      Impedance Z+ (phi z)
        (.3D0,0.D0)      Impedance Z- (phi z)
        (.3D0,0.D0)      Impedance Z+ (z phi)
```

-
~~~~~~

```
        (.3D0,0.D0)      Impedance Z- (z phi)
        (.4D0,0.D0)      Impedance Z+ (z z)
        (.5D0,0.D0)      Impedance Z- (z z)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

C. SETUP PROGRAM AND CREATED FILES

```
      PROGRAM SETUP
C***********************************************************************
C NOTES:The format statement 1001 need to be revised if other than
C        double precision real numbers are used for DKHMAX and DKAMAX.
C        Statement 1001 needs to be revised if KHMAX or KAMAX exceeds
C        3 digits.
C       The format statement 1002 need to be revised if other than
C       double precision real numbers are used for FZERO AND PRECSN.
C       Statement 1002 needs to be revised if I0BIT exceeds 6 digits
C       or IFPBIT exceeds 4 digits.
C***********************************************************************
C
      INCLUDE 'REALTP.INC'
      INCLUDE 'CMPXTP.INC'
      OPEN (20,FILE='CYLGEOM.PRM',IOSTAT=IOS,STATUS='OLD')
         IF(IOS .NE. 0) THEN
      WRITE(*,9000)
9000  FORMAT ('Cannot find the file CYLGEOM.PRM containing the ',/,
     +  'maximum values for ka and kh, and the parameter NREGNS.')
      STOP
         END IF
      READ (20,*) KHMAX
      READ (20,*) KAMAX
      READ (20,*) NREGNS
      CLOSE (20)
      OPEN (20,FILE='CYLFLPT.PRM',IOSTAT=IOS,STATUS='OLD')
         IF(IOS .NE. 0) THEN
      WRITE(*,9001)
9001  FORMAT ('Cannot find the file CYLFLPT.PRM containing floating',/,
     +  ' point zero bit I0BIT and precision IFPBIT.')
      STOP
         END IF
      READ(20,*) I0BIT
      READ(20,*) IFPBIT
      CLOSE (20)
      OPEN (21,FILE='LIMITS.INC',STATUS='UNKNOWN')
      WRITE (21,1001) KHMAX, KAMAX
      WRITE (21,1002) I0BIT, IFPBIT, IFPBIT
      CLOSE (21)
1001  FORMAT (6X,'PARAMETER (DKHMAX= ',I3,'.D0, DKAMAX= ',I3,'.D0)')
1002  FORMAT (6X,'PARAMETER (FZERO=',I6,'.D0, PRECSN=',I4,'.D0,',
     +        ' IFPBIT=',I4,')')
C
C  Part 2
      DKHMAX=KHMAX
      DKAMAX=KAMAX
      ONEDEL=ONE-EPS8
      KQDIM=INT((DKHMAX/PI)*NREGNS+ONEDEL)
      KNDIM=INT((DKAMAX/TWO)*NREGNS+ONEDEL)
      KQDIM=MAX(KQDIM,1)
      KNDIM=MAX(KNDIM,1)
      OPEN (21,FILE='MAINDM.INC',STATUS='UNKNOWN')
      WRITE (21,1003) NREGNS, KNDIM, KQDIM
      WRITE (21,1004)
      WRITE (21,1005)
      WRIE (21,1006)
      CLOSE (21)
1003  FORMAT (6X,'PARAMETER (NREGNS=',I2,', KNDIM=',I3,
     +        ', KQDIM=',I3,')')
```

```
1004  FORMAT (6X,'PARAMETER (KNDIM1=KNDIM+1, KQDIM1=KQDIM+1)')
1005  FORMAT (6X,'PARAMETER (KXCRT=2*KQDIM1, KCRNT=4*KQDIM1)')
1006  FORMAT (6X,'PARAMETER (MAXNG=KNDIM1, MAXPEG=KQDIM/2+1,',
     +        ' MAXPOG=KQDIM1/2)')
C
C  Part 3
      FZERO=IOBIT
      REFC=FZERO*LOG(TWO)-LOG(PI2)-ONE
      REFH=HALF*(REFC-LOG(DKHMAX))
      REFA=HALF*(REFC-LOG(DKAMAX))
C
      KQ2=KQDIM+2
      DMXM=KQ2
         DO 100 WHILE ((DMXM+HALF)*(LOG(DMXM/DKHMAX)+ONEN) .LT. REFH)
      DMXM=DMXM+ONE
100      CONTINUE
      MXMREG=INT(DMXM)
C
      MXMSNG=INT(TWO*DKHMAX+ONEDEL)
      MXMSNG=MAX(IFPBIT,MXMSNG,KQ2)+KNDIM+1
      OPEN (21,FILE='GPQNDM.INC',STATUS='UNKNOWN')
      WRITE (21,1009) MXMREG, MXMSNG
      CLOSE (21)
1009  FORMAT (6X,'PARAMETER (MXMREG=',I4,', MXMSNG=',I4,')')
      STOP
      END
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
GPQNDM.INC
       PARAMETER (DKHMAX=  30.D0, DKAMAX=   5.D0)
      PARAMETER (FZERO=  1024.D0, PRECSN=  64.D0, IFPBIT=  64)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
LIMITS.INC
      PARAMETER (DKHMAX=  30.D0, DKAMAX=   5.D0)
      PARAMETER (FZERO=  1024.D0, PRECSN=  64.D0, IFPBIT=  64)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
MAINDM.INC
      PARAMETER (NREGNS= 8, KNDIM= 20, KQDIM= 77)
      PARAMETER (KNDIM1=KNDIM+1, KQDIM1=KQDIM+1)
      PARAMETER (KXCRT=2*KQDIM1, KCRNT=4*KQDIM1)
      PARAMETER (MAXNG=KNDIM1, MAXPEG=KQDIM/2+1, MAXPOG=KQDIM1/2)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

D. PROGRAM MAIN

```
      PROGRAM MAIN
**********************************************************************
      INCLUDE 'REALTP.INC'
      INCLUDE 'CMPXTP.INC'
      COMMON /GCONST/ DKH,DKA,HH,HA,HSQ,DASQ,HSQN,DASQN,HHSQ,HDASQ,
     +          RAH,RAHSQ,DHA,DAH
      COMMON /INPUT1/ DKA0,DINCA,NSTR,NEND,RHA
      COMMON /INPUT2/ IE,IM,THETAI,THESINI,THECOSI,RTHEI
      COMMON /INPUT4/ IZ,IK,IS,NYSM
C
      CALL CHKINPUT
      OPEN (21,FILE='rhzz41.dz',STATUS='UNKNOWN')
      DO 8001 IS=NSTR,NEND
      DS=IS
      DKA=DKA0+DINCA*DS
      DKH=DKA*RHA
      CALL MAXODM
      CALL RAPQMT
      CALL GNPQFN
      CALL BCKSCFL(IR,CESTH,CESPH)
```

```
      CALL RCSPAREA(CESTH,CESPH)
8001  CONTINUE
      CLOSE(21)
      STOP
      END
```

E. SUBROUTINE CHKINPUT

```
      SUBROUTINE CHKINPUT
C****************************************************************************
      INCLUDE 'REALTP.INC'
      INCLUDE 'LIMITS.INC'
      COMMON /INPUT1/ DKA0,DINCA,NSTR,NEND,RHA
      COMMON /INPUT2/ IE,IM,THETAI,THESINI,THECOSI,RTHEI
      COMMON /INPUT3/ RTHE,RDELT,RPHI,RDELP,NTHTAO,NPHI,THESIN,THECOS,
     +                RHPI
      COMMON /INPUT4/ IZ,IK,IS,NYSM
C**********************************************************************
      OPEN(20,FILE='INPUTDAT.PRM',IOSTAT=IOS,STATUS='OLD')
         IF(IOS .NE. 0) THEN
      WRITE(*,*) 'Fail to open input file INPUTDAT.PRM'
      STOP
         END IF
C**********************************************************************
C  Input kh and ka. These values are passed to other
C  parts of this program through the common block /INPUT1/.
      READ(20,*) DKA0
      READ(20,*) DINCA
      READ(20,*) NSTR
      READ(20,*) NEND
      READ(20,*) RHA
C**********************************************************************
C  Check against maximum kh and ka values.
      NBTW=NEND-NSTR
      DKH0=DKA0*RHA
      DKA1=DINCA*NBTW+DKA0
      DKH1=DKA1*RHA
        IF (DKH1 .GT. DKHMAX) THEN
            IF (DKA1 .GT. DKAMAX) THEN
      WRITE(*,*) 'Both kh and ka values exceed the maximum allowed.'
            ELSE
      WRITE(*,*) 'The input kh value exceeds the maximum value allowed.'
            END IF
      WRITE(*,*) 'The execution is terminated.'
      CLOSE(20)
      STOP
           ELSE IF (DKA1 .GT. DKAMAX)THEN
      WRITE(*,*) 'The input ka value exceeds the maximum value allowed.'
      WRITE(*,*) 'The execution is terminated.'
      CLOSE(20)
      STOP
         END IF
        IF ((DINCH .LT. ZERO) .OR. (DINCA .LT. ZERO)) THEN
      WRITE(*,*) 'The increment DINCH or DINCA is less than zero.'
      WRITE(*,*) 'The execution is terminated.'
      CLOSE(20)
      STOP
         END IF
C**********************************************************************
C  Input the incident angle (THETAI) and polarization (TE or TM) of the
C  incident wave. The incident angle is limited to 0 to 90
C  degrees. SIN(THETAI) and COS(THETAI) are computed also. These
C  parameters are passed to other parts of this program through the
C  common block /INPUT2/.
      READ(20,*) IE
      READ(20,*) IM
      READ(20,*) THETAI
```

```
C  Check the input value of incident wave
        IF ((IE .NE. 1) .AND. (IE. NE. 0)) THEN
      WRITE (*,*) 'Improperly specified the polarization of incident ',
     +  'wave. program is stoped.'
      STOP
        ELSE IF ((IM .NE. 1) .AND. (IM .NE. 0)) THEN
      WRITE (*,*) 'Improperly specified the polarization of incident ',
     +  'wave. program is stopped.'
      CLOSE(20)
      STOP
        END IF

        IF ((THETAI .LT. ZERO) .OR. (THETAI .GT. 90.)) THEN
      WRITE (*,*) 'Improperly specified incident angle. ',
     +  'program is stopped.'
      CLOSE(20)
      STOP
        END IF
C Calculate SIN(THETAI) and COS(THETAI)
      RTHEI=THETAI*PI/DEGPI
      THESINI=SIN(RTHEI)
      THECOSI=COS(RTHEI)
C*********************************************************************
C  Input the angles theta and phi at which the scattered fields are
C  to be computed. They are specified in terms of the initial theta
C  (THETAO) and phi (PHIO) angles, their respective increments DELTHO
C  and DELPHI, and the total numbers of angles NTHTAO and NPHI to be
C  computed. Thus NTHTAO and NPHI must be integers greater than 1. If
C  either NTHTAO=0 or NPHI=0, no bistatically scattered fields will be
C  computed. Note that the scattered electric field components are
C  computed for all the phi-angles at a fixed theta, before the
C  theta-angle is varied. All angles are specified in degrees. Theta is
C  limited to 0 to 180 while phi is limited to 0 to 360 degrees.
C  SIN(THETAO) and COS(THETAO) are computed also.  These parameters are
C  passed to other parts of this program through the common block /INPUT3/.
      READ(20,*) NTHTAO
      READ(20,*) NPHI
        IF ((NTHTAO .LT. 0) .OR. (NPHI .LT. 0)) THEN
      WRITE(*,*) 'Improperly specified number of output angles. ',
     +  'Program is stopped.'
      STOP
        ELSE IF ((NTHTAO .EQ. 0) .OR. (NPHI .EQ. 0)) THEN
      CLOSE(20)
      WRITE(*,*) 'Desired bistatic scattered field direction has not ',
     +  'been (properly) specified, they will not be computed.'
      NTHTAO=0
      NPHI=0
      THETAO=ZERO
      DELTHO=ZERO
      PHIO=ZERO
      DELPHI=ZERO
        ELSE
      READ(20,*) THETAO
      READ(20,*) DELTHO
      READ(20,*) PHIA
      READ(20,*) DELPHI
         END IF
      READ(20,*) IK
C*********************************************************************
C  Check input values.
        IF ((DKH0 .LE. ZERO) .OR. (DKH1 .LE. ZERO))THEN
      WRITE(*,*) 'Invalid kh, program is stopped.'
      STOP
        END IF
        IF ((DKA0 .LE. ZERO) .OR. (DKA1 .LE. ZERO))THEN
      WRITE(*,*) 'Invalid ka, program is stopped.'
      STOP
        END IF
```

```
         IF ((DKA0 .GT. DKH0) .OR. (DKA1 .GT. DKH1))THEN
      WRITE(*,*) 'ka/kh > 1, program is stopped.'
      STOP
         END IF
C  Output angle checking not required:
         IF (NTHTAO .EQ. 0) GO TO 200
C  Chcking output angles:
         IF ((THETAO .LT. ZERO) .OR. (THETAO .GT. DEGPI)) THEN
      WRITE(*,*) 'The first output theta-angle lies outside the 0 to  ',
     +  '180 degrees range. Program is stopped.'
      STOP
         END IF
      THTAIF=THETAO+(NTHTAO-1)*DELTHO
         IF ((THTAIF .LT. ZERO) .OR. (THTAIF .GT. DEGPI)) THEN
      WRITE(*,*) 'Some of the specified output theta-angles lie',
     +  'outside the 0 to 180 degrees range. Program is stopped.'
      STOP
         END IF
      PHIMX=TWO*DEGPI
         IF ((PHIO .LT. ZERO) .OR. (PHIO .GT. PHIMX)) THEN
      WRITE(*,*) 'The first output phi-angle lies outside the 0 to ',
     +  '360 degrees range. Program is stopped.'
      STOP
         END IF
      PHIF=PHIO+(NPHI-1)*DELPHI
         IF ((PHIF .LT. ZERO) .OR. (PHIF .GT. TWO*DEGPI)) THEN
      WRITE(*,*) 'Some of the output phi-angles lie outside',
     +  'the 0 to 360 degrees range. Program is stopped.'
      STOP
         END IF
200   CONTINUE
      DPI=PI/DEGPI
      RTHE=THETAO*DPI
      RPHI=PHIO*DPI
      RDELT=DELTHO*DPI
      RDELP=DELPHI*DPI
      RHPI=90.*DPI
      THESIN=SIN(RTHE)
      THECOS=COS(RTHE)
      RETURN
      END
```

F. SUBROUTINE **MAXODM**

```
      SUBROUTINE MAXODM
C *********************************************************************
      INCLUDE 'REALTP.INC'
      INCLUDE 'MAINDM.INC'
      COMMON /CRNTDM/ NMAX,MXNG,IQMAX,IQMAX1,IQMAX2,IXCRNT,ICRNT,MXQEG,
     +          MXQOG
      COMMON /GCONST/ DKH,DKA,HH,HA,HSQ,DASQ,HSQN,DASQN,HHSQ,HDASQ,
     +         RAH,RAHSQ,DHA,DAH
      COMMON /RTHETA/ DL1COSI,DL2SINI,DLL
      COMMON /INPUT1/ DKA0,DINCA,NSTR,NEND,RHA
      COMMON /INPUT2/ IE,IM,THETAI,THESINI,THECOSI,RTHEI
      COMMON /INPUT3/ RTHE,RDELT,RPHI,RDELP,NTHTAO,NPHI,THESIN,THECOS,
     +                RHPI
C
      ONEDEL=ONE-EPS8
      IQMAX=INT((DKH/PI)*NREGNS+ONEDEL)
      IQMAX=MAX(IQMAX,1)
      IQMAX1=IQMAX+1
      IQMAX2=IQMAX1+2
      IXCRNT=2*IQMAX1
      ICRNT=4*IQMAX1
      MXQEG=IQMAX/2+1
      MXQOG=IQMAX1/2
```

```
C
      NMAX=INT((DKA/TWO)*NREGNS+ONEDEL)
      NMAX=MAX(NMAX,1)
      MXNG=NMAX+1
C
C  Evaluate SIN and COS functions to pass along /
      DL1COSI=DKH*THECOSI
      DL2SINI=DKA*THESINI
      DLL=DKH*DL2SINI
C
C  Evaluate geometrical values to pass along /GCONST/.
      HH=HALF*DKH
      HA=HALF*DKA
      HSQ=DKH*DKH
      DASQ=DKA*DKA
      HSQN=-HSQ
      DASQN=-DASQ
      HHSQ=QUAR*HSQ
      HDASQ=QUAR*DASQ
      RAH=DKA/DKH
      RAHSQ=RAH*RAH
      DHA=DKH*HA
      DAH=HH*HA
C
      RETURN
      END
```

G. SUBROUTINE RAPQMT

```
      SUBROUTINE RAPQMT
C*********************************************************************
      INCLUDE 'REALTP.INC'
      INCLUDE 'CMPXTP.INC'
      INCLUDE 'MAINDM.INC'
      DIMENSION CO(2,2), CIR(2,2)
      DIMENSION CZSUM(2,2),CZDIF(2,2),CR1(4,4),CR2(4,4),CR3(4,4)
      DIMENSION CRPQ1(KCRNT,KCRNT),CRPQ2(KCRNT,KCRNT),
     +          CRPQ31(KCRNT,KCRNT),CRPQ32(KCRNT,KCRNT)
      COMMON /INPUT2/ IE,IM,THETAI,THESINI,THECOSI,RTHEI
      COMMON /INPUT4/ IZ,IK,IS,NYSM
      COMMON /XPQTMP1/ CRPQ1,CRPQ2
      COMMON /XPQTMP2/ CRPQ31,CRPQ32
      COMMON /CRNTDM/ NMAX,MXNG,IQMAX,IQMAX1,IQMAX2,IXCRNT,ICRNT,MXQEG,
     +                MXQOG
C
      OPEN (20,FILE='IMPEDNCE.PRM',IOSTAT=IOS,STATUS='OLD')
         IF (IOS .NE. 0) THEN
      WRITE(*,9000)
9000  FORMAT ('Cannot find the file IMPEDNCE1.PRM containing the ',/,
     +  'impedances of the inner and outer surfaces.')
      STOP
         END IF
C
      READ(20,*) IZ
         IF ((IZ .NE. 0) .AND. (IZ .NE. 1)) THEN
      WRITE (*,*) 'Improperly specified the impedance of cylinder ',
     +  'surface. program is stopped.'
      CLOSE(20)
      STOP
         END IF
         IF (IZ .EQ. 0) THEN
      CLOSE(20)
      RETURN
         END IF
C Set up the matrix when Z and Delta are diagonal, or not diagonal but n  ≤0.
C Initialize the matrix CRPQ1
      DO 200 I=1,KCRNT
```

```
          DO 100 J=1,KCRNT
       CRPQ1(I,J)=CZERO
       CRPQ2(I,J)=CZERO
       CRPQ31(I,J)=CZERO
       CRPQ32(I,J)=CZERO
100       CONTINUE
200    CONTINUE
       DO 400 I=1,2
          DO 300 J=1,2
       READ (20,*) CO(I,J)
       READ (20,*) CIR(I,J)
       CZSUM(I,J)=(CO(I,J)+CIR(I,J))*HALF
       CZDIF(I,J)=(CO(I,J)-CIR(I,J))*HALF
300       CONTINUE
400    CONTINUE
       CLOSE (20)
       CZ11=CZSUM(1,1)
       CZ12=CZSUM(1,2)
       CZ21=CZSUM(2,1)
       CZ22=CZSUM(2,2)
       CRDET=CZ11*CZ22-CZ12*CZ21
       CD11=CZDIF(1,1)
       CD12=CZDIF(1,2)
       CD21=CZDIF(2,1)
       CD22=CZDIF(2,2)
          IF (CRDET .EQ. CZERO) THEN
       WRITE (*,9001)
       STOP
           END IF
C Check the diagonalization of the impedance matrixes
       NSYM=1
          IF (CZ12 .NE. CZERO) THEN
       NSYM=0
          ELSE IF (CZ21 .NE. CZERO) THEN
       NSYM=0
          ELSE IF (CD12 .NE. CZERO) THEN
       NSYM=0
          ELSE IF (CD21 .NE. CZERO) THEN
       NSYM=0
          END IF
       CRDTZ=CONE/CRDET
       CR1(1,1)=CZ11-(CD11*CD11*CZ22-CD11*CD12*CZ21-CD11*CD21*CZ12
     +          +CD12*CD21*CZ11)*CRDTZ
       CR1(1,2)=CZ12-(CD11*CD12*CZ22-CD12*CD12*CZ21-CD11*CD22*CZ12
     +          +CD12*CD22*CZ11)*CRDTZ
       CR1(2,1)=CZ21-(CD11*CD21*CZ22-CD11*CD22*CZ21-CD21*CD21*CZ12
     +          +CD21*CD22*CZ11)*CRDTZ
       CR1(2,2)=CZ22-(CD12*CD21*CZ22-CD12*CD22*CZ21-CD21*CD22*CZ12
     +          +CD22*CD22*CZ11)*CRDTZ
       CR1(1,3)=(CD12*CZ11-CD11*CZ12)*CRDTZ
       CR1(1,4)=(CD12*CZ21-CD11*CZ22)*CRDTZ
       CR1(2,3)=(CD22*CZ11-CD21*CZ12)*CRDTZ
       CR1(2,4)=(CD22*CZ21-CD21*CZ22)*CRDTZ
       CR1(3,1)=(CD11*CZ21-CD21*CZ11)*CRDTZ
       CR1(3,2)=(CD12*CZ21-CD22*CZ11)*CRDTZ
       CR1(4,1)=(CD11*CZ22-CD21*CZ12)*CRDTZ
       CR1(4,2)=(CD12*CZ22-CD22*CZ12)*CRDTZ
       CR1(3,3)=CZ11*CRDTZ
       CR1(3,4)=CZ21*CRDTZ
       CR1(4,3)=CZ12*CRDTZ
       CR1(4,4)=CZ22*CRDTZ
C
       DO 1300 IQE=0,MXQEG
       IQE1=IQE+1
       IQ=2*IQE
       DQ=IQ
       DQ1=IQ+1
       IQX1=4*IQ+1
```

```
      IQX2=IQX1+1
      IQX3=IQX2+1
      IQX4=IQX3+1
         DO 1100 IPE=0,MXQEG
      IPE1=IPE+1
      IP=2*IPE
      IPX1=4*IP+1
      IPX2=IPX1+1
      IPX3=IPX2+1
      IPX4=IPX3+1
      DP=IP
      DP1=IP+1
      DBPHI=PISQ*(DP+DQ1)*(DP1-DQ)
      BPHI=4.D0/DBPHI
      DBZ=PISQ*(DP+DQ1)*(DP1-DQ)*(DP1+DQ1+ONE)*(DP-DQ1)
      BZ=-8.D0*DQ1/DBZ
      CRPQ1(IPX1,IQX1)=CR1(1,1)*BPHI
      CRPQ1(IPX2,IQX1)=CR1(2,1)*BPHI
      CRPQ1(IPX3,IQX1)=CR1(3,1)*BPHI
      CRPQ1(IPX4,IQX1)=CR1(4,1)*BPHI
      CRPQ1(IPX1,IQX2)=CR1(1,2)*BZ
      CRPQ1(IPX2,IQX2)=CR1(2,2)*BZ
      CRPQ1(IPX3,IQX2)=CR1(3,2)*BZ
      CRPQ1(IPX4,IQX2)=CR1(4,2)*BZ
      CRPQ1(IPX1,IQX3)=CR1(1,3)*BPHI
      CRPQ1(IPX2,IQX3)=CR1(2,3)*BPHI
      CRPQ1(IPX3,IQX3)=CR1(3,3)*BPHI
      CRPQ1(IPX4,IQX3)=CR1(4,3)*BPHI
      CRPQ1(IPX1,IQX4)=CR1(1,4)*BZ
      CRPQ1(IPX2,IQX4)=CR1(2,4)*BZ
      CRPQ1(IPX3,IQX4)=CR1(3,4)*BZ
      CRPQ1(IPX4,IQX4)=CR1(4,4)*BZ
1100     CONTINUE
1300  CONTINUE
      DO 2300 IQO=0,MXQOG
      IQO1=IQO+1
      IQ=2*IQO+1
      IQX1=4*IQ+1
      IQX2=IQX1+1
      IQX3=IQX2+1
      IQX4=IQX3+1
      DQ=IQ
      DQ1=IQ+1
         DO 2200 IPO=0,MXQOG
      IPO1=IPO+1
      IP=2*IPO+1
      IPX1=4*IP+1
      IPX2=IPX1+1
      IPX3=IPX2+1
      IPX4=IPX3+1
      DP=IP
      DP1=IP+1
      DBPHI=PISQ*(DP+DQ1)*(DP1-DQ)
      BPHI=4.D0/DBPHI
      DBZ=PISQ*(DP+DQ1)*(DP1-DQ)*(DP1+DQ1+ONE)*(DP-DQ1)
      BZ=-8.D0*DQ1/DBZ
      CRPQ1(IPX1,IQX1)=CR1(1,1)*BPHI
      CRPQ1(IPX2,IQX1)=CR1(2,1)*BPHI
      CRPQ1(IPX3,IQX1)=CR1(3,1)*BPHI
      CRPQ1(IPX4,IQX1)=CR1(4,1)*BPHI
      CRPQ1(IPX1,IQX2)=CR1(1,2)*BZ
      CRPQ1(IPX2,IQX2)=CR1(2,2)*BZ
      CRPQ1(IPX3,IQX2)=CR1(3,2)*BZ
      CRPQ1(IPX4,IQX2)=CR1(4,2)*BZ
      CRPQ1(IPX1,IQX3)=CR1(1,3)*BPHI
      CRPQ1(IPX2,IQX3)=CR1(2,3)*BPHI
      CRPQ1(IPX3,IQX3)=CR1(3,3)*BPHI
      CRPQ1(IPX4,IQX3)=CR1(4,3)*BPHI
```

```
      CRPQ1(IPX1,IQX4)=CR1(1,4)*BZ
      CRPQ1(IPX2,IQX4)=CR1(2,4)*BZ
      CRPQ1(IPX3,IQX4)=CR1(3,4)*BZ
      CRPQ1(IPX4,IQX4)=CR1(4,4)*BZ
2200     CONTINUE
2300  CONTINUE
C Set up the matrix Rpq utilized when Z or Delta is nor diagonal, and n < 0.
      CR2(1,2)=-CR1(1,2)
      CR2(2,1)=-CR1(2,1)
      CR2(1,3)=-CR1(1,3)
      CR2(2,4)=-CR1(2,4)
      CR2(3,1)=-CR1(3,1)
      CR2(4,2)=-CR1(4,2)
      CR2(3,4)=-CR1(3,4)
      CR2(4,3)=-CR1(4,3)
      CR2(1,1)=CR1(1,1)
      CR2(2,2)=CR1(2,2)
      CR2(1,4)=CR1(1,4)
      CR2(2,3)=CR1(2,3)
      CR2(3,2)=CR1(3,2)
      CR2(4,1)=CR1(4,1)
      CR2(3,3)=CR1(3,3)
      CR2(4,4)=CR1(4,4)
      DO 3300 IQE=0,MXQEG
      IQE1=IQE+1
      IQ=2*IQE
      DQ=IQ
      DQ1=IQ+1
      IQX1=4*IQ+1
      IQX2=IQX1+1
      IQX3=IQX2+1
      IQX4=IQX3+1
         DO 3100 IPE=0,MXQEG
      IPE1=IPE+1
      IP=2*IPE
      IPX1=4*IP+1
      IPX2=IPX1+1
      IPX3=IPX2+1
      IPX4=IPX3+1
      DP=IP
      DP1=IP+1
      DBPHI=PISQ*(DP+DQ1)*(DP1-DQ)
      BPHI=4.D0/DBPHI
      DBZ=PISQ*(DP+DQ1)*(DP1-DQ)*(DP1+DQ1+ONE)*(DP-DQ1)
      BZ=-8.D0*DQ1/DBZ
      CRPQ2(IPX1,IQX1)=CR2(1,1)*BPHI
      CRPQ2(IPX2,IQX1)=CR2(2,1)*BPHI
      CRPQ2(IPX3,IQX1)=CR2(3,1)*BPHI
      CRPQ2(IPX4,IQX1)=CR2(4,1)*BPHI
      CRPQ2(IPX1,IQX2)=CR2(1,2)*BZ
      CRPQ2(IPX2,IQX2)=CR2(2,2)*BZ
      CRPQ2(IPX3,IQX2)=CR2(3,2)*BZ
      CRPQ2(IPX4,IQX2)=CR2(4,2)*BZ
      CRPQ2(IPX1,IQX3)=CR2(1,3)*BPHI
      CRPQ2(IPX2,IQX3)=CR2(2,3)*BPHI
      CRPQ2(IPX3,IQX3)=CR2(3,3)*BPHI
      CRPQ2(IPX4,IQX3)=CR2(4,3)*BPHI
      CRPQ2(IPX1,IQX4)=CR2(1,4)*BZ
      CRPQ2(IPX2,IQX4)=CR2(2,4)*BZ
      CRPQ2(IPX3,IQX4)=CR2(3,4)*BZ
      CRPQ2(IPX4,IQX4)=CR2(4,4)*BZ
3100     CONTINUE
3300  CONTINUE
      DO 4300 IQO=0,MXQOG
      IQO1=IQO+1
      IQ=2*IQO+1
      IQX1=4*IQ+1
      IQX2=IQX1+1
```

```
      IQX3=IQX2+1
      IQX4=IQX3+1
      DQ=IQ
      DQ1=IQ+1
         DO 4200 IPO=0,MXQOG
      IPO1=IPO+1
      IP=2*IPO+1
      IPX1=4*IP+1
      IPX2=IPX1+1
      IPX3=IPX2+1
      IPX4=IPX3+1
      DP=IP
      DP1=IP+1
      DBPHI=PISQ*(DP+DQ1)*(DP1-DQ)
      BPHI=4/DBPHI
      DBZ=PISQ*(DP+DQ1)*(DP1-DQ)*(DP1+DQ1+ONE)*(DP-DQ1)
      BZ=-8.*DQ1/DBZ
      CRPQ2(IPX1,IQX1)=CR2(1,1)*BPHI
      CRPQ2(IPX2,IQX1)=CR2(2,1)*BPHI
      CRPQ2(IPX3,IQX1)=CR2(3,1)*BPHI
      CRPQ2(IPX4,IQX1)=CR2(4,1)*BPHI
      CRPQ2(IPX1,IQX2)=CR2(1,2)*BZ
      CRPQ2(IPX2,IQX2)=CR2(2,2)*BZ
      CRPQ2(IPX3,IQX2)=CR2(3,2)*BZ
      CRPQ2(IPX4,IQX2)=CR2(4,2)*BZ
      CRPQ2(IPX1,IQX3)=CR2(1,3)*BPHI
      CRPQ2(IPX2,IQX3)=CR2(2,3)*BPHI
      CRPQ2(IPX3,IQX3)=CR2(3,3)*BPHI
      CRPQ2(IPX4,IQX3)=CR2(4,3)*BPHI
      CRPQ2(IPX1,IQX4)=CR2(1,4)*BZ
      CRPQ2(IPX2,IQX4)=CR2(2,4)*BZ
      CRPQ2(IPX3,IQX4)=CR2(3,4)*BZ
      CRPQ2(IPX4,IQX4)=CR2(4,4)*BZ
4200     CONTINUE
4300  CONTINUE
C  Prepare a mtarix for computing the scattering currents on the outer
C  and the inner surfaces
         IF (IK .NE. 1) THEN
      RETURN
         END IF
      CR3(1,1)=-CR1(4,1)
      CR3(1,2)=-CR1(4,2)
      CR3(1,3)=-CR1(4,3)
      CR3(1,4)=-CR1(4,4)
      CR3(3,1)=CR1(2,1)
      CR3(3,2)=CR1(2,2)
      CR3(3,3)=CR1(2,3)
      CR3(3,4)=CR1(2,4)
      CR3(2,1)=CR1(3,1)
      CR3(2,2)=CR1(3,2)
      CR3(2,3)=CR1(3,3)
      CR3(2,4)=CR1(3,4)
      CR3(4,1)=-CR1(1,1)
      CR3(4,2)=-CR1(1,2)
      CR3(4,3)=-CR1(1,3)
      CR3(4,4)=-CR1(1,4)
C
      DO 5200 IQE=0,MXQEG
      IQE1=IQE+1
      IQ=2*IQE
      DQ=IQ
      DQ1=IQ+1
      IQX1=4*IQ+1
      IQX2=IQX1+1
      IQX3=IQX2+1
      IQX4=IQX3+1
         DO 5100 IPE=0,MXQEG
      IPE1=IPE+1
```

```
      IP=2*IPE
      IPX1=4*IP+1
      IPX2=IPX1+1
      IPX3=IPX2+1
      IPX4=IPX3+1
      DP=IP
      DP1=IP+1
      DBPHI=PISQ*(DP+DQ1)*(DP1-DQ)
      BPHI=4./DBPHI
      DBZ=PISQ*(DP+DQ1)*(DP1-DQ)*(DP1+DQ1+ONE)*(DP-DQ1)
      BZ=-8.*DQ1/DBZ
      CRPQ31(IPX1,IQX1)=(CONE+CR3(1,1))*BPHI
      CRPQ31(IPX2,IQX1)=CR3(2,1)*BPHI
      CRPQ31(IPX3,IQX1)=CR3(3,1)*BPHI
      CRPQ31(IPX4,IQX1)=CR3(4,1)*BPHI
      CRPQ31(IPX1,IQX2)=CR3(1,2)*BZ
      CRPQ31(IPX2,IQX2)=(CONE+CR3(2,2))*BZ
      CRPQ31(IPX3,IQX2)=CR3(3,2)*BZ
      CRPQ31(IPX4,IQX2)=CR3(4,2)*BZ
      CRPQ31(IPX1,IQX3)=CR3(1,3)*BPHI
      CRPQ31(IPX2,IQX3)=CR3(2,3)*BPHI
      CRPQ31(IPX3,IQX3)=(CONE+CR3(3,3))*BPHI
      CRPQ31(IPX4,IQX3)=CR3(4,3)*BPHI
      CRPQ31(IPX1,IQX4)=CR3(1,4)*BZ
      CRPQ31(IPX2,IQX4)=CR3(2,4)*BZ
      CRPQ31(IPX3,IQX4)=CR3(3,4)*BZ
      CRPQ31(IPX4,IQX4)=(CONE+CR3(4,4))*BZ
C
      CRPQ32(IPX1,IQX1)=(CONE-CR3(1,1))*BPHI
      CRPQ32(IPX2,IQX1)=-CR3(2,1)*BPHI
      CRPQ32(IPX3,IQX1)=-CR3(3,1)*BPHI
      CRPQ32(IPX4,IQX1)=-CR3(4,1)*BPHI
      CRPQ32(IPX1,IQX2)=-CR3(1,2)*BZ
      CRPQ32(IPX2,IQX2)=(CONE-CR3(2,2))*BZ
      CRPQ32(IPX3,IQX2)=-CR3(3,2)*BZ
      CRPQ32(IPX4,IQX2)=-CR3(4,2)*BZ
      CRPQ32(IPX1,IQX3)=-CR3(1,3)*BPHI
      CRPQ32(IPX2,IQX3)=-CR3(2,3)*BPHI
      CRPQ32(IPX3,IQX3)=(CONE-CR3(3,3))*BPHI
      CRPQ32(IPX4,IQX3)=-CR3(4,3)*BPHI
      CRPQ32(IPX1,IQX4)=-CR3(1,4)*BZ
      CRPQ32(IPX2,IQX4)=-CR3(2,4)*BZ
      CRPQ32(IPX3,IQX4)=-CR3(3,4)*BZ
      CRPQ32(IPX4,IQX4)=(CONE-CR3(4,4))*BZ
5100     CONTINUE
5200  CONTINUE
C
      DO 6200 IQO=0,MXQOG
      IQO1=IQO+1
      IQ=2*IQO+1
      DQ=IQ
      DQ1=IQ+1
      IQX1=4*IQ+1
      IQX2=IQX1+1
      IQX3=IQX2+1
      IQX4=IQX3+1
         DO 6100 IPO=0,MXQOG
      IPE1=IPO+1
      IP=2*IPO+1
      IPX1=4*IP+1
      IPX2=IPX1+1
      IPX3=IPX2+1
      IPX4=IPX3+1
      DP=IP
      DP1=IP+1
      DBPHI=PISQ*(DP+DQ1)*(DP1-DQ)
      BPHI=4./DBPHI
      DBZ=PISQ*(DP+DQ1)*(DP1-DQ)*(DP1+DQ1+ONE)*(DP-DQ1)
```

```
      BZ=-8.*DQ1/DBZ
      CRPQ31(IPX1,IQX1)=(CONE+CR3(1,1))*BPHI
      CRPQ31(IPX2,IQX1)=CR3(2,1)*BPHI
      CRPQ31(IPX3,IQX1)=CR3(3,1)*BPHI
      CRPQ31(IPX4,IQX1)=CR3(4,1)*BPHI
      CRPQ31(IPX1,IQX2)=CR3(1,2)*BZ
      CRPQ31(IPX2,IQX2)=(CONE+CR3(2,2))*BZ
      CRPQ31(IPX3,IQX2)=CR3(3,2)*BZ
      CRPQ31(IPX4,IQX2)=CR3(4,2)*BZ
      CRPQ31(IPX1,IQX3)=CR3(1,3)*BPHI
      CRPQ31(IPX2,IQX3)=CR3(2,3)*BPHI
      CRPQ31(IPX3,IQX3)=(CONE+CR3(3,3))*BPHI
      CRPQ31(IPX4,IQX3)=CR3(4,3)*BPHI
      CRPQ31(IPX1,IQX4)=CR3(1,4)*BZ
      CRPQ31(IPX2,IQX4)=CR3(2,4)*BZ
      CRPQ31(IPX3,IQX4)=CR3(3,4)*BZ
      CRPQ31(IPX4,IQX4)=(CONE+CR3(4,4))*BZ
C
      CRPQ32(IPX1,IQX1)=(CONE-CR3(1,1))*BPHI
      CRPQ32(IPX2,IQX1)=-CR3(2,1)*BPHI
      CRPQ32(IPX3,IQX1)=-CR3(3,1)*BPHI
      CRPQ32(IPX4,IQX1)=-CR3(4,1)*BPHI
      CRPQ32(IPX1,IQX2)=-CR3(1,2)*BZ
      CRPQ32(IPX2,IQX2)=(CONE-CR3(2,2))*BZ
      CRPQ32(IPX3,IQX2)=-CR3(3,2)*BZ
      CRPQ32(IPX4,IQX2)=-CR3(4,2)*BZ
      CRPQ32(IPX1,IQX3)=-CR3(1,3)*BPHI
      CRPQ32(IPX2,IQX3)=-CR3(2,3)*BPHI
      CRPQ32(IPX3,IQX3)=(CONE-CR3(3,3))*BPHI
      CRPQ32(IPX4,IQX3)=-CR3(4,3)*BPHI
      CRPQ32(IPX1,IQX4)=-CR3(1,4)*BZ
      CRPQ32(IPX2,IQX4)=-CR3(2,4)*BZ
      CRPQ32(IPX3,IQX4)=-CR3(3,4)*BZ
      CRPQ32(IPX4,IQX4)=(CONE-CR3(4,4))*BZ
6100     CONTINUE
6200  CONTINUE
      RETURN
9001  FORMAT('The given inside and outside impedance have a singular',
     +       ' sum. Execution is terminated.')
      END
```

H. SUBROUTINE GNPQFN

```
       SUBROUTINE GNPQFN
C****************************************************************************
      INCLUDE 'REALTP.INC'
      INCLUDE 'MAINDM.INC'
      COMMON /GCONST/ DKH,DKA,HH,HA,HSQ,DASQ,HSQN,DASQN,HHSQ,HDASQ,
     +         RAH,RAHSQ,DHA,DAH
      CHARACTER FILEEVEN1*12, FILEODD1*12
C Set up the file names
      NC1=INT(DKA)
      DC=NC1
      NC=100000.*DKH+NC1
      DC1=DKA-DC
      NC2=INT(1000.*DC1)
C Set up the indices of file names of Green's function, and check whether
C these files exist. If these files exist, then use it directly. Otherwise,
C call subroutine XPQINI1.
      FILEEVEN1='E       .   '
      FILEODD1='O       .   '
      IGE=8*2*(MAXPEG+1)*(MAXPEG+2)
      IGO=8*2*(MAXPOG+1)*(MAXPOG+2)
      WRITE (FILEEVEN1(2:8),'(I7.7)') NC
      WRITE (FILEODD1(2:8),'(I7.7)') NC
      WRITE (FILEEVEN1(10:12),'(I3.3)') NC2
```

```
      WRITE (FILEODD1(10:12),'(I3.3)') NC2
      OPEN (28,ACCESS='DIRECT',FILE=FILEEVEN1,RECL=IGE,IOSTAT=IOS,
     +         STATUS='OLD')
         IF (IOS .NE. 0) THEN
      CLOSE (28)
      CALL XPQINI1(DKH,DKA)
          ELSE
      OPEN (29,ACCESS='DIRECT',FILE=FILEODD1,RECL=IGO,IOSTAT=IOS,
     +         STATUS='OLD')
            IF (IOS .NE. 0) THEN
      CLOSE (29)
      CALL XPQINI1(DKH,DKA)
            ELSE
      CLOSE (28)
      CLOSE (29)
      CALL XPQINI(DKH,DKA)
            END IF
         ENDIF
      RETURN
      END
C
      SUBROUTINE XPQINI(DKHIN,DKAIN)
C***********************************************************************
      INCLUDE 'REALTP.INC'
      INCLUDE 'CMPXTP.INC'
      INCLUDE 'MAINDM.INC'
      DIMENSION GNE(4,(MAXPEG+1)*(MAXPEG+2)/2),
     +          GNO(4,(MAXPOG+1)*(MAXPOG+2)/2)
      DIMENSION CGNE(0:MAXPEG,0:MAXPEG,KNDIM1+1),
     +          CGNO(0:MAXPOG,0:MAXPOG,KNDIM1+1)
      DIMENSION CDGNE(0:MAXPEG,0:MAXPEG,KNDIM1+1),
     +          CDGNO(0:MAXPOG,0:MAXPOG,KNDIM1+1)
      COMMON /CRNTDM/ NMAX,MXNG,IQMAX,IQMAX1,IQMAX2,IXCRNT,ICRNT,MXQEG,
     +                MXQOG
      COMMON /GPQTMP/ CGNE,CDGNE,CGNO,CDGNO
      SAVE /GPQTMP/
C
      CHARACTER FILEEVEN1*12, FILEODD1*12
      DKH=DKHIN
      DKA=DKAIN
c
C Initialize the matrix CGNE, CGNO, CDGNE, CDGNO
      DO 105 IC= 1,KNDIM1+1
           DO 102 IB= 0,MAXPEG
              DO 101 IA= 0,MXPEG
      CGNE(IA,IB,IC)=CZERO
      CDGNE(IA,IB,IC)=CZERO
101           CONTINUE
102        CONTINUE
           DO 104 ID=0,MAXPOG
              DO 103 IE=0,MAXPOG
      CGNO(IE,ID,IC)=CZERO
      CDGNO(IE,ID,IC)=CZERO
103         CONTINUE
104      CONTINUE
105   CONTINUE
C
C Set up the file name
      NC1=INT(DKA)
      DC=NC1
      NC=100000.*DKH+NC1
      DC1=DKA-DC
      NC2=INT(1000.*DC1)
      FILEEVEN1='E       .   '
      FILEODD1='O        .   '
      IGE=8*2*(MAXPEG+1)*(MAXPEG+2)
      IGO=8*2*(MAXPOG+1)*(MAXPOG+2)
      WRITE (FILEEVEN1(2:8),'(I7.7)') NC
```

```
      WRITE (FILEODD1(2:8),'(I7.7)') NC
      WRITE (FILEEVEN1(10:12),'(I3.3)') NC2
      WRITE (FILEODD1(10:12),'(I3.3)') NC2
      OPEN (28,ACCESS='DIRECT',FILE=FILEEVEN1,RECL=IGE,IOSTAT=IOS,
     +         STATUS='OLD')
C
      OPEN (29,ACCESS='DIRECT',FILE=FILEODD1,RECL=IGO,IOSTAT=IOS,
     +         STATUS='OLD')

      DO 900 NI=1,MXNG+1
C     The following values N, DN and DNH are passed to the G-computation
C     related subroutines through the common block /NCONST/.
      READ (28,REC=NI) GNE
      IRECE=0
         DO 500 IQE=0,MXQEG
            DO 300 IPE=0,IQE-1
      CGNE(IPE,IQE,NI)=CGNE(IQE,IPE,NI)
      CDGNE(IPE,IQE,NI)=CDGNE(IQE,IPE,NI)
300         CONTINUE
            DO 400 IPE=IQE,MXQEG
      IRECE=IRECE+1
      GR=GNE(1,IRECE)
      GI=GNE(2,IRECE)
      GDR=GNE(3,IRECE)
      GDI=GNE(4,IRECE)
      CGNE(IPE,IQE,NI)=DCMPLX(GR,GI)
      CDGNE(IPE,IQE,NI)=DCMPLX(GDR,GDI)
400         CONTINUE
500      CONTINUE
      READ (29,REC=NI) GNO
      IRECO=0
         DO 800 IQO=0,MXQOG
            DO 600 IPO=0,IQO-1
      CGNO(IPO,IQO,NI)=CGNO(IQO,IPO,NI)
      CDGNO(IPO,IQO,NI)=CDGNO(IQO,IPO,NI)
600         CONTINUE
            DO 700 IPO=IQO,MXQOG
      IRECO=IRECO+1
      GR=GNO(1,IRECO)
      GI=GNO(2,IRECO)
      GDR=GNO(3,IRECO)
      GDI=GNO(4,IRECO)
      CGNO(IPO,IQO,NI)=DCMPLX(GR,GI)
      CDGNO(IPO,IQO,NI)=DCMPLX(GDR,GDI)
700         CONTINUE
800      CONTINUE
900   CONTINUE
      CLOSE (28)
      CLOSE (29)
      RETURN
      END

C
      SUBROUTINE XPQINI1(DKHIN,DKAIN)
C*********************************************************************
      INCLUDE 'REALTP.INC'
      INCLUDE 'CMPXTP.INC'
      INCLUDE 'MAINDM.INC'
      INCLUDE 'GPQNDM.INC'
      INCLUDE 'LIMITS.INC'
      DIMENSION GNE(4,(MAXPEG+1)*(MAXPEG+2)/2),
     +           GNO(4,(MAXPOG+1)*(MAXPOG+2)/2)
      DIMENSION CGNE(0:MAXPEG,0:MAXPEG,KNDIM1+1),
     +           CGNO(0:MAXPOG,0:MAXPOG,KNDIM1+1)
      DIMENSION CDGNE(0:MAXPEG,0:MAXPEG,KNDIM1+1),
     +           CDGNO(0:MAXPOG,0:MAXPOG,KNDIM1+1)
      COMMON /CRNTDM/ NMAX,MXNG,IQMAX,IQMAX1,IQMAX2,IXCRNT,ICRNT,MXQEG,
     +                MXQOG
```

```
      COMMON /GCONST/ DKH,DKA,HH,HA,HSQ,DASQ,HSQN,DASQN,HHSQ,HDASQ,
     +                RAH,RAHSQ,DHA,DAH
      COMMON /SUMLMT/ DHMAX,DAMAX,DSNG,IHMAX,IAMAX,ISNG
      COMMON /NCONST/ DN,DNH,N
      COMMON /GPQTMP/ CGNE,CDGNE,CGNO,CDGNO
      CHARACTER FILEEVEN1*12, FILEODD1*12
      SAVE /GPQTMP/,/GCONST/,/SUMLMT/
C
      DKH=DKHIN
      DKA=DKAIN
C Initialize the matrix CGNE, CGNO, CDGNE, CDGNO
      DO 105 IC= 1,KNDIM1+1
           DO 102 IB= 0,MAXPEG
              DO 101 IA= 0,MXPEG
      CGNE(IA,IB,IC)=CZERO
      CDGNE(IA,IB,IC)=CZERO
101           CONTINUE
102        CONTINUE
           DO 104 ID=0,MAXPOG
              DO 103 IE=0,MAXPOG
      CGNO(IE,ID,IC)=CZERO
      CDGNO(IE,ID,IC)=CZERO
103         CONTINUE
104     CONTINUE
105   CONTINUE
C
C  Determine the maximum number of terms for r- and m- series sums and
C  pass them through /SUMLMT/:
C
      REFC=FZERO*LOG(TWO)-LOG(PI2)-ONE
      REFH=HALF*(REFC-LOG(DKH))
      REFA=HALF*(REFC-LOG(DKA))
C
      IQMX2=IQMAX+2
      DHMAX=IQMX2
         DO 1100 WHILE ((DHMAX+HALF)*(LOG(DHMAX/DKH)+ONEN) .LT. REFH)
      DHMAX=DHMAX+ONE
1100     CONTINUE
      IHMAX=INT(DHMAX)
C
      DAMAX=AINT(DKA)+ONE
         DO 1200 WHILE ((DAMAX+HALF)*(LOG(DAMAX/DKA)+ONEN) .LT. REFA)
      DAMAX=DAMAX+ONE
1200     CONTINUE
C
      IAMAX=INT(DAMAX)
C
      ISNG=INT(TWO*DKH+ONE-EPS8)
      ISNG=MAX(IFPBIT,ISNG,IQMX2)
C  Checking dimensions.
         IF(IHMAX .GT. MXMREG) THEN
      WRITE(*,*) 'Warning: IHMAX = ',IHMAX,' > MXMREG = ',MXMREG
      WRITE(*,*) 'IHMAX IS SET TO MXMREG IN XPQINI1'
      IHMAX=MXMREG
         END IF
      ISN1=MXMSNG-N-1
         IF(ISNG .GT. ISN1) THEN
      WRITE(*,*) 'Warning: ISNG = ',ISNG,' > MXMSNG-N-1 = ',ISN1
      WRITE(*,*) 'ISNG IS REDUCED TO MXMSNG-N-1 IN XPQINI1'
      ISNG=ISN1
         END IF
C
      DSNG=ISNG
C
      FILEEVEN1='E       .   '
      FILEODD1='O       .   '
      NC1=INT(DKA)
      DC=NC1
```

```
      NC=DKH*100000.+NC1
      DC1=DKA-DC
      NC2=INT(1000.*DC1)
      IGE=8*2*(MAXPEG+1)*(MAXPEG+2)
      IGO=8*2*(MAXPOG+1)*(MAXPOG+2)
      WRITE (FILEEVEN1(2:8),'(I7.7)') NC
      WRITE (FILEODD1(2:8),'(I7.7)') NC
      WRITE (FILEEVEN1(10:12),'(I3.3)') NC2
      WRITE (FILEODD1(10:12),'(I3.3)') NC2
      OPEN (28,ACCESS='DIRECT',FILE=FILEEVEN1,RECL=IGE,STATUS='UNKNOWN')
      OPEN (29,ACCESS='DIRECT',FILE=FILEODD1,RECL=IGO,STATUS='UNKNOWN')
C  Initialize CGNE, CDGNE, CGNO and CDGNO
      DO 1900 NI=1,MXNG+1
C     The following values N, DN and DNH are passed to the G-computation
C     related subroutines through the common block /NCONST/.
      N=NI-1
      DN=N
      DNH=DN+HALF
      IRECE=0
         DO 1500 IQE=0,MXQEG
      IQ=2*IQE
            DO 1300 IPE=0,IQE-1
      IP=2*IPE
      CGNE(IPE,IQE,NI)=CGNE(IQE,IPE,NI)
      CDGNE(IPE,IQE,NI)=CDGNE(IQE,IPE,NI)
1300        CONTINUE
            DO 1400 IPE=IQE,MXQEG
      IRECE=IRECE+1
      IP=2*IPE
      CALL GDN(IP,IQ,GI,GDI,GR,GDR)
      GNE(1,IRECE)=GR
      GNE(2,IRECE)=GI
      GNE(3,IRECE)=GDR
      GNE(4,IRECE)=GDI
      CGNE(IPE,IQE,NI)=DCMPLX(GR,GI)
      CDGNE(IPE,IQE,NI)=DCMPLX(GDR,GDI)
1400        CONTINUE
1500     CONTINUE
      WRITE (28,REC=NI) GNE
      IRECO=0
         DO 1800 IQO=0,MXQOG
      IQ=2*IQO+1
            DO 1600 IPO=0,IQO-1
      IP=2*IPO+1
      CGNO(IPO,IQO,NI)=CGNO(IQO,IPO,NI)
      CDGNO(IPO,IQO,NI)=CDGNO(IQO,IPO,NI)
1600        CONTINUE
            DO 1700 IPO=IQO,MXQOG
      IRECO=IRECO+1
      IP=2*IPO+1
      CALL GDN(IP,IQ,GI,GDI,GR,GDR)
      GNO(1,IRECO)=GR
      GNO(2,IRECO)=GI
      GNO(3,IRECO)=GDR
      GNO(4,IRECO)=GDI
      CGNO(IPO,IQO,NI)=DCMPLX(GR,GI)
      CDGNO(IPO,IQO,NI)=DCMPLX(GDR,GDI)
1700        CONTINUE
1800     CONTINUE
      WRITE (29,REC=NI) GNO
1900  CONTINUE
      CLOSE(28)
      CLOSE(29)
      RETURN
      END
C
      SUBROUTINE GDN(IPIN,IQIN,GIOUT,GDIOUT,GROUT,GDROUT)
C***********************************************************************
```

```
C  This subrouitne sets up P and Q dependent constants and passes along
C  /PCONST/, then calls subroutines GDREG and GDSNG to compute G(P,Q,N)
C  and its derivative.
C**********************************************************************
C
      INCLUDE 'REALTP.INC'
      COMMON /PCONST/ DP,DQ,DS,DD,DSH,DDH,DSSQ,DDSQ,IP,IQ,IS,ID
C  Check and transform input variables.
      IP=IPIN
      IQ=IQIN
      ISPQ=IP+IQ
      IS=ISPQ/2
         IF ((ISPQ+1)/2 .GT. IS) THEN
      GIOUT=ZERO
      GDIOUT=ZERO
      GROUT=ZERO
      GDROUT=ZERO
      WRITE (*, 9001) ISPQ
9001  FORMAT('The parameters P and Q have a sum of the ODD integer ',I4,
     +/,', G(P, Q, N) and its derivative have been set to 0.')
      RETURN
         END IF
         IF (IP .LT. IQ) THEN
      IP=IQIN
      IQ=IPIN
         END IF
C
      DP=IP
      DQ=IQ
C
      ID=IS-IQ
      DS=IS
      DD=ID
      DSH=DS+HALF
      DDH=DD+HALF
      DSSQ=DS*DS
      DDSQ=DD*DD
C
      CALL GDREG(GI,GDI,GRR,GDRR)
      CALL GDSNG(GRS,GDRS)
      GIOUT=GI
      GDIOUT=GDI
      GROUT=GRR+GRS
      GDROUT=GDRR+GDRS
      RETURN
      END
CC
      SUBROUTINE GDREG(GIOUT,GDIOUT,GROUT,GDROUT)
C**********************************************************************
C  This subrouitne computes the regular part of G(P,Q,N) and its
C  derivative.
C**********************************************************************
C
      INCLUDE 'REALTP.INC'
        INCLUDE 'GPQNDM.INC'
      COMMON /GCONST/ DKH,DKA,HH,HA,HSQ,DASQ,HSQN,DASQN,HHSQ,HDASQ,
     +                RAH,RAHSQ,DHA,DAH
      COMMON /NCONST/ DN,DNH,N
      COMMON /PCONST/ DP,DQ,DS,DD,DSH,DDH,DSSQ,DDSQ,IP,IQ,IS,ID
      COMMON /SUMLMT/ DHMAX,DAMAX,DSNG,IHMAX,IAMAX,ISNG
      SAVE /GCONST/,/SUMLMT/
C
C  Reserve working space to store r-independent numbers.
      DIMENSION GIM(MXMREG),GRRM(MXMREG)
C
C  Computation starts.
C
C  Compute overal constant factors:
```

```
      GRRF=QUAR*DKH/PISQ/(DSSQ-QUAR)/(QUAR-DDSQ)/(DN+ONE)
      GIF=HALF/DSH
      DJN=ZERO
         DO 300 JN=1,N
      DJN=DJN+ONE
      HATN=HA/DJN
      GRRF=GRRF*HATN*HATN
      GIF=GIF*HATN*HA/(DJN+DSH)
300      CONTINUE
      DJP=ZERO
         DO 400 JQ=1,IQ
      DJP=DJP+ONE
      GIF=(HHSQ/DJP/DJP)*GIF
400      CONTINUE
         DO 500 JP=IQ+1,IP
      DJP=DJP+ONE
      GIF=(HH/DJP)*GIF
500      CONTINUE
      IF ((ID+1)/2 .GT. ID/2) GIF=-GIF
C
C  Compute GI, GDI, GRREG, and GDRREG.
C     Compute r-independent factors and store in GIM and GRRM:
      SMH=DSH+ONEN
      SM1=DS
      SM2=DS+DS
      PM1=DP
      QM1=DQ
      THRH=HALF
      DM1=ZERO
      THRS=HALF+DS
      THRD=HALF+DD
      THRDN=HALF-DD
      THRSN=HALF-DS
         DO 600 JM=1,IHMAX
      SMH=SMH+ONE
      SM1=SM1+ONE
      SM2=SM2+ONE
      PM1=PM1+ONE
      QM1=QM1+ONE
      THRH=THRH+ONE
      DM1=DM1+ONE
      THRS=THRS+ONE
      THRD=THRD+ONE
      THRDN=THRDN+ONE
      THRSN=THRSN+ONE
      GIM(JM)=(SMH/SM2)*(SMH/PM1)*(SM1/QM1)*(HSQN/DM1)
      GRRM(JM)=(THRH/THRS)*(HSQN/THRD)*(DM1/THRDN)*(DM1/THRSN)
600      CONTINUE
C
C     Compute r- and m-sum for GI and GRR.
C        Setup initial r related values
      DJR=DAMAX
      DNR=DN+DJR
      DNHR=DNR-HALF
      D2NR=DN+DNR
      DNSHR=DNR+DSH
      DN2R=DNR+ONE
C           m-sum for r=IAMAX
      DNSHRM=DNSHR+DHMAX
      DN2RM=DN2R+DHMAX
      ANNEXTR=ONE+GIM(IHMAX)/DNSHRM
      ANR0R=ONE+GRRM(IHMAX)/DN2RM
         DO 700 JM=IHMAX-1,1,-1
      DNSHRM=DNSHRM+ONEN
      DN2RM=DN2RM+ONEN
      ANNEXTR=ONE+ANNEXTR*GIM(JM)/DNSHRM
      ANR0R=ONE+ANR0R*GRRM(JM)/DN2RM
700      CONTINUE
```

```
      ANNEXT=ANNEXTR
      DANNEXT=DNR*ANNEXTR
      ANR0=ANR0R
      DANR0=DNR*ANR0R
         DO 900 JR=IAMAX,1,-1
C        Compute factors for r=JR
      ATMP=(DNHR/D2NR)*(DASQN/DJR)
      ATMPI=ATMP/DNSHR
      ATMPR=ATMP/DN2R
C        Setup new r related values for r=JR-1
      DJR=DJR+ONEN
      DNR=DNR+ONEN
      DNHR=DNHR+ONEN
      D2NR=D2NR+ONEN
      DNSHR=DNSHR+ONEN
      DN2R=DN2R+ONEN
C        m-sum for r=JR-1
      DNSHRM=DNSHR+DHMAX
      DN2RM=DN2R+DHMAX
      ANNEXTR=ONE+GIM(IHMAX)/DNSHRM
      ANR0R=ONE+GRRM(IHMAX)/DN2RM
         DO 800 JM=IHMAX-1,1,-1
      DNSHRM=DNSHRM+ONEN
      DN2RM=DN2RM+ONEN
      ANNEXTR=ONE+ANNEXTR*GIM(JM)/DNSHRM
      ANR0R=ONE+ANR0R*GRRM(JM)/DN2RM
800      CONTINUE
      ANNEXT=ANNEXTR+ANNEXT*ATMPI
      DANNEXT=DNR*ANNEXTR+DANNEXT*ATMPI
      ANR0=ANR0R+ANR0*ATMPR
      DANR0=DNR*ANR0R+DANR0*ATMPR
900      CONTINUE
C
      GIOUT=ANNEXT*GIF
      GDIOUT=DANNEXT*GIF
      GROUT=ANR0*GRRF
      GDROUT=DANR0*GRRF
C
      RETURN
      END
C
      SUBROUTINE GDSNG(GROUT,GDROUT)
C**********************************************************************
C  This subrouitne computes the 'singular' part of G(P,Q,N) and its
C  derivative.
C**********************************************************************
C
      INCLUDE 'REALTP.INC'
      INCLUDE 'GPQNDM.INC'
      COMMON /GCONST/ DKH,DKA,HH,HA,HSQ,DASQ,HSQN,DASQN,HHSQ,HDASQ,
     +               RAH,RAHSQ,DHA,DAH
      COMMON /NCONST/ DN,DNH,N
      COMMON /PCONST/ DP,DQ,DS,DD,DSH,DDH,DSSQ,DDSQ,IP,IQ,IS,ID
      COMMON /SUMLMT/ DHMAX,DAMAX,DSNG,IHMAX,IAMAX,ISNG
      SAVE /GCONST/,/SUMLMT/
C
C  Reserve working space to store r-independent numbers.
      DIMENSION GS1M(MXMSNG),GS2M(MXMSNG),
     +          GR2M(MXMSNG),GR2R(0:MXMSNG),GDR2R(0:MXMSNG)
C
C  Computation starts.
C
      MXGR2M=ISNG+N
c
C  Compute overal constant factors:
      GRSF=QUAR/PISQ/DKH
C
C  Compute GR1, GDR1, GR2, and GDR2:
```

```
C     Compute initial constants:

      S1LHA=TWO*LOG(HXD/RAH)
      DK=HALF
      S1SD0=ZERO
         DO 1000 JK=1,ID
      S1SD0=S1SD0+ONE/DK
      DK=DK+ONE
1000     CONTINUE
      S1SD0=TWO*S1SD0
         DO 1100 JK=ID+1, IS
      S1SD0=S1SD0+ONE/DK
      DK=DK+ONE
1100     CONTINUE
      S1SD0=-TWO*S1SD0
      SN0N=ZERO
      SN10=ZERO
      SN20=ZERO
         IF (N .GT. 0) THEN
      DJN=ONE
      DJNH=HALF
            DO 1200 JN=1,N-1
      DJNI=ONE/DJN
      DJNHI=ONE/DJNH
      SN10=DJNI*DJNHI+SN10
      SN20=DJNHI*DJNHI*((ONE-QUAR*DJNI)*HALF*DJNI+ONE)+SN20
      DJN=DJN+ONE
      DJNH=DJNH+ONE
1200        CONTINUE
      DJNI=ONE/DJN
      DJNHI=ONE/DJNH
      SN0N=S1LHA+HALF*DJNHI-SN10*QUAR
      SN10=(DJNI*DJNHI+SN10)*QUAR
      SN20=HALF*(DJNHI*DJNHI*((QUAR*DJNI+ONEN)*HALF*DJNI+ONEN)-SN20)
         END IF
      SN10=S1LHA-SN10
      SN20=PISQ*THIR+SN20
C
C     Compute r-independent terms and store in GR2M, GS1M, and GS2M:
      GR2JM=ONE
      S1SDM=S1SD0
      S2SDM=ZERO
      DM=ZERO
      DMH=-HALF
         DO 1300 JM=1,MXGR2M
      DM=DM+ONE
      DMH=DMH+ONE
      DMI=ONE/DM
      DMHI=ONE/DMH
      DMHSQ=DMH*DMH
      DMHSS=DMHSQ-DSSQ
      DMHSD=DMHSQ-DDSQ
      GR2JM=(DMHSS*DMI*DMI)*(DMHSD*DMI*DMHI)*GR2JM
      GR2M(JM)=GR2JM
      TWSMHS=TWO*DSSQ/DMHSS
      TWDMHD=TWO*DDSQ/DMHSD
      S1SDM=S1SDM-(TWSMHS+TWDMHD+DMI+ONE)*DMHI
      S2SDM=S2SDM-((TWSMHS+THR)*TWSMHS+(TWDMHD+THR)*TWDMHD+
     +       (ONE-QUAR*DMI)*TWO*DMI+ONE)/DMHSQ
      GS1M(JM)=S1SDM
      GS2M(JM)=S2SDM
1300     CONTINUE
C
C     Compute m-sum for GR1, GDR1 and store in GR2R, GDR2R
         IF (N .GT. 0) THEN
      DJR=DN+ONEN
      DJRH=DJR-HALF
      SN0R=SN0N
```

```
      SGR1M=SN0R+S1SD0
      SDGR1M=DJR*SGR1M+ONEN
      DJM=ONE
      DJRM=DJR
            DO 1400 JM=1,N-1
      SR1M=SN0R+GS1M(JM)
      HSQF=HSQ/DJM/DJRM
      SGR1M=SGR1M*HSQF+SR1M*GR2M(JM)
      SDGR1M=SDGR1M*HSQF+(DJR*SR1M+ONEN)*GR2M(JM)
      DJM=DJM+ONE
      DJRM=DJRM+ONEN
1400        CONTINUE
      GR2R(N-1)=SGR1M*TWO
      GDR2R(N-1)=SDGR1M*TWO
            DO 1600 JR=N-2,0,-1
      SN0R=SN0R+ONE/DJRH+ONE/(DN-DJR)-ONE/(DN+DJR)
      DJR=DJR+ONEN
      DJRH=DJRH+ONEN
      SGR1M=SN0R+S1SD0
      SDGR1M=DJR*SGR1M+ONEN
      DJM=ONE
      DJRM=DJR
               DO 1500 JM=1,JR
      SR1M=SN0R+GS1M(JM)
      HSQF=HSQ/DJM/DJRM
      SGR1M=SGR1M*HSQF+SR1M*GR2M(JM)
      SDGR1M=SDGR1M*HSQF+(DJR*SR1M+ONEN)*GR2M(JM)
      DJM=DJM+ONE
      DJRM=DJRM+ONEN
1500           CONTINUE
      GR2R(JR)=SGR1M*TWO
      GDR2R(JR)=SDGR1M*TWO
1600        CONTINUE
         END IF
C
C     Compute m-sum for GR2, GDR2 and store in GR2R, GDR2R
      DRN=DN
      SN1R=SN10
      SN2R=SN20
      SR2MM=SN1R+S1SD0
      SR2RMS=SR2MM*SR2MM+SN2R
      DR2RMS=DRN*SR2RMS-TWO*SR2MM
          DJM=ONE
      DJRM=DRN
         DO 1700 JM=1,N
      SR2MM=SN1R+GS1M(JM)
      SR2RM=SR2MM*SR2MM+SN2R+GS2M(JM)
      DR2RM=DRN*SR2RM-TWO*SR2MM
      HSQF=HSQ/DJM/DJRM
      SR2RMS=SR2RMS*HSQF+SR2RM*GR2M(JM)
      DR2RMS=DR2RMS*HSQF+DR2RM*GR2M(JM)
      DJM=DJM+ONE
      DJRM=DJRM+ONEN
1700  CONTINUE
      GR2R(N)=SR2RMS
      GDR2R(N)=DR2RMS
      DJR=ZERO
         DO 1900 JR=N+1,MXGR2M
      DJR=DJR+ONE
      DRN=DRN+ONE
      DJRI=ONE/DJR
      DRNHI=ONE/(DRN-HALF)
      DR2NI=ONE/(DRN+DN)
      DNHRF=DNH*DRNHI*DR2NI
      SN1R=DJRI-DNHRF+SN1R
      SN2R=DJRI*DJRI-(DRNHI+DR2NI)*DNHRF+SN2R
      SR2MM=SN1R+S1SD0
      SR2RMS=SR2MM*SR2MM+SN2R
```

```
      DR2RMS=DRN*SR2RMS-TWO*SR2MM
      DJM=ONE
      DJRM=DRN
            DO 1800 JM=1,JR
      SR2MM=SN1R+GS1M(JM)
      SR2RM=SR2MM*SR2MM+SN2R+GS2M(JM)
      DR2RM=DRN*SR2RM-TWO*SR2MM
      HSQF=HSQ/DJM/DJRM
      SR2RMS=SR2RMS*HSQF+SR2RM*GR2M(JM)
      DR2RMS=DR2RMS*HSQF+DR2RM*GR2M(JM)
      DJM=DJM+ONE
      DJRM=DJRM+ONEN
1800        CONTINUE
      GR2R(JR)=SR2RMS
      GDR2R(JR)=DR2RMS
1900     CONTINUE
C
C        Compute r-sum:
C
         IF (RAHSQ .GT. HALF) THEN
C
C           Convergence acceleration via Euler-Abel Transfomation:
C
      DJR=ZERO
      DJRH=DJR-HALF
      DJRN=DJR+DN
      DJRNN=DJR-DN
            IF (N .GT. 0) THEN
      RFCTOR=ONE/DN
      GR2R(0)=RFCTOR*GR2R(0)
      GDR2R(0)=RFCTOR*GDR2R(0)
               DO 2000 JR=1,N-1
      DJR=DJR+ONE
      DJRH=DJRH+ONE
      DJRN=DJRN+ONE
      DJRNN=DJRNN+ONE
      RFCTOR=(DJR*DJRH/DJRN/DJRNN)*RAHSQ*RFCTOR
      GR2R(JR)=RFCTOR*GR2R(JR)
      GDR2R(JR)=RFCTOR*GDR2R(JR)
2000           CONTINUE
      DJR=DJR+ONE
      DJRH=DJRH+ONE
      DJRN=DJRN+ONE
      DJRNN=DJRNN+ONE
      RFCTOR=HALF*(HALF-DN)*RAHSQ*RFCTOR
      GR2R(N)=RFCTOR*GR2R(N)
      GDR2R(N)=RFCTOR*GDR2R(N)
            ELSE
      RFCTOR=ONE
            END IF
            DO 2100 JR=N+1,MXGR2M
      DJR=DJR+ONE
      DJRH=DJRH+ONE
      DJRN=DJRN+ONE
      DJRNN=DJRNN+ONE
      RFCTOR=(DJR*DJRH/DJRN/DJRNN)*RAHSQ*RFCTOR
      GR2R(JR)=RFCTOR*GR2R(JR)
      GDR2R(JR)=RFCTOR*GDR2R(JR)
2100        CONTINUE
C
C              Compute zeroth-order delta-r term:
            DO 2300 JDELTR=1,MXGR2M
               DO 2200 JR=MXGR2M,JDELTR,-1
      GR2R(JR)=GR2R(JR-1)-GR2R(JR)
      GDR2R(JR)=GDR2R(JR-1)-GDR2R(JR)
2200           CONTINUE
2300        CONTINUE
      SGR2R=HALF*GR2R(MXGR2M)
```

```
      SDGR2R=HALF*GDR2R(MXGR2M)
            DO 2400 JR=MXGR2M-1,0,-1
      SGR2R=(SGR2R+GR2R(JR))*HALF
      SDGR2R=(SDGR2R+GDR2R(JR))*HALF
2400        CONTINUE
      GROUT=GRSF*SGR2R
      GDROUT=GRSF*SDGR2R
C
         ELSE
C
C           Direct summation when RAHSQ is no greater than 1/2:
C
      DJR=DSNG
      DJRN=DJR+DN
      DJRNH=DJRN-HALF
      DJR2N=DJRN+DN
      RFCTOR=(DJRN*DJRNH/DJR/DJR2N)*RAHSQ
      SGR2R=RFCTOR*GR2R(MXGR2M)
      SDGR2R=RFCTOR*GDR2R(MXGR2M)
            DO 2500 JR=MXGR2M-1,N+1,-1
      DJR=DJR+ONEN
      DJRN=DJRN+ONEN
      DJRNH=DJRNH+ONEN
      DJR2N=DJR2N+ONEN
      RFCTOR=(DJRN*DJRNH/DJR/DJR2N)*RAHSQ
      SGR2R=(GR2R(JR)-SGR2R)*RFCTOR
      SDGR2R=(GDR2R(JR)-SDGR2R)*RFCTOR
2500        CONTINUE
            IF (N .EQ. 0) THEN
      GROUT=(GR2R(0)-SGR2R)*GRSF
      GDROUT=(GDR2R(0)-SDGR2R)*GRSF
            ELSE
      RFCTOR=HALF*(HALF-DN)*RAHSQ
      SGR2R=(GR2R(N)-SGR2R)*RFCTOR
      SDGR2R=(GDR2R(N)-SDGR2R)*RFCTOR
      DJR=DN
      DJRH=DJR-HALF
      DJRN=DJR+DN
      DJRNN=DJR-DN
               DO 2600 JR=N-1,1,-1
      DJR=DJR+ONEN
      DJRH=DJRH+ONEN
      DJRN=DJRN+ONEN
      DJRNN=DJRNN+ONEN
      RFCTOR=(DJR*DJRH/DJRN/DJRNN)*RAHSQ
      SGR2R=(GR2R(JR)-SGR2R)*RFCTOR
      SDGR2R=(GDR2R(JR)-SDGR2R)*RFCTOR
2600           CONTINUE
      GROUT=(GR2R(0)-SGR2R)*GRSF/DN
      GDROUT=(GDR2R(0)-SDGR2R)*GRSF/DN
            END IF
         END IF
      RETURN
      END
C
```

I. SUBROUTINE BCKSCFL

```
        SUBROUTINE BCKSCFL(IR,CESTH,CESPH)
C****************************************************************************
C
      INCLUDE 'REALTP.INC'
      INCLUDE 'CMPXTP.INC'
      INCLUDE 'MAINDM.INC'
C
      REAL*4 AKPHPR,AKPHPI,AKZPR,AKZPI,ALPHPR,ALPHPI,ALZPR,ALZPI,
```

```
     +       AKPHNR,AKPHNI,AKZNR,AKZNI,ALPHNR,ALPHNI,ALZNR,ALZNI
      DIMENSION CIW(KCRNT),CKLN(KCRNT)
      DIMENSION CIW0(KXCRT),CKLN0(KXCRT),CXPQN0(KXCRT,KXCRT)
      DIMENSION CRPQ1(KCRNT,KCRNT),CRPQ2(KCRNT,KCRNT),
     +          CXPQN(KCRNT,KCRNT),CXRPQ(KCRNT,KCRNT)
      DIMENSION CKPHP(61,61),CKZP(61,61),CLPHP(61,61),
     +          CLZP(61,61), CKPHN(61,61),CKZN(61,61),
     +          CLPHN(61,61),CLZN(61,61)
      COMMON /INPUT2/ IE,IM,THETAI,THESINI,THECOSI,RTHEI
      COMMON /INPUT4/ IZ,IK,IS,NYSM
      COMMON /NCONST/ DN,DNH,N
      COMMON /XPQTMP/ CXPQN,CXRPQ,CXPQN0
      COMMON /XPQTMP1/ CRPQ1,CRPQ2
      COMMON /CKLMTX/ CKPHP,CKZP,CLPHP,CLZP,CKPHN,CKZN,CLPHN,CLZN
      COMMON /CRNTDM/ NMAX,MXNG,IQMAX,IQMAX1,IQMAX2,IXCRNT,ICRNT,MXQEG,
     +                MXQOG
      COMMON /INPUT3/ RTHE,RDELT,RPHI,RDELP,NTHTAO,NPHI,THESIN,THECOS,
     +                RHPI
      CESTH=CZERO
      CESPH=CZERO
      DO 100 JQ=1,KXCRT
      CKLN0(JQ)=CZERO
100   CONTINUE
      DO 200 JQ=1,KCRNT
      CKLN(JQ)=CZERO
200   CONTINUE
      DO 400 IF=-30,30
         DO 300 JZ=-30,30
      CKPHP(IF,JZ)=CZERO
      CKZP(IF,JZ)=CZERO
      CLPHP(IF,JZ)=CZERO
      CLZP(IF,JZ)=CZERO
      CKPHN(IF,JZ)=CZERO
      CKZN(IF,JZ)=CZERO
      CLPHN(IF,JZ)=CZERO
      CLZN(IF,JZ)=CZERO
300      CONTINUE
400   CONTINUE
C
        IF (RTHEI .EQ. 0.D0) GO TO 2000
C If incident angle is not equal to 0, use this loop
      DO 1800 NA=0,NMAX
C
      CALL INCIDNT(NA,CIW0,CIW)
C
         IF (NA .EQ. 0) THEN
      CALL XPQ0
         ELSE
      CALL XPQN(NA)
         ENDIF
C If the cylinder is made of perfect conductor and no coating on it
          IF (IZ .EQ. 0) THEN
      DN=NA
C Use IMSL library to solve linear system
      CALL DLSACG(IXCRNT,CXPQN0,KXCRT,CIW0,1,CKLN0)
      CALL ESCFAR(NA,CKLN0,CKLN,CETHN,CEPHN)
C
C  If we calculate in X and Y components as theta approaches 0 or pi,
C  then theta component is cosin phi in X direction plus sine phi in Y
C  direction, and phi component is negative sine phi in X direction plus
C  cosin phi in Y direction.
         IF (RTHE .EQ. 0.D0)  THEN
      CETHN=COS(RPHI)*CETHN+SIN(RPHI)*CEPHN
      CEPHN=-SIN(RPHI)*CETHN+COS(RPHI)*CEPHN
         ELSEIF (RTHE .EQ. PI) THEN
      CETHN=-COS(RPHI)*CETHN-SIN(RPHI)*CEPHN
      CEPHN=-SIN(RPHI)*CETHN+COS(RPHI)*CEPHN
          END IF
```

```
C
      CESTH=CESTH+CETHN
      CESPH=CESPH+CEPHN
            IF (NA .NE. 0) THEN
               IF (IE .EQ. 1) THEN
      DO 600 I=2,KXCRT,2
      CKLN0(I)=-CKLN0(I)
600   CONTINUE
               END IF
               IF (IM .EQ. 1) THEN
      DO 800 I=1,KXCRT,2
      CKLN0(I)=-CKLN0(I)
800   CONTINUE
               END IF
      NA1=-NA
      DN=NA1
      CALL ESCFAR(NA1,CKLN0,CKLN,CETHN,CEPHN)
            IF (RTHE .EQ. 0.D0)  THEN
      CETHN=COS(RPHI)*CETHN+SIN(RPHI)*CEPHN
      CEPHN=-SIN(RPHI)*CETHN+COS(RPHI)*CEPHN
            ELSEIF (RTHE .EQ. PI) THEN
      CETHN=-COS(RPHI)*CETHN-SIN(RPHI)*CEPHN
      CEPHN=-SIN(RPHI)*CETHN+COS(RPHI)*CEPHN
             END IF
C
      CESTH=CESTH+CETHN
      CESPH=CESPH+CEPHN
            END IF
         END IF
C If the cylinder is coated with anisotropic material
         IF (IZ .EQ. 1) THEN
      DO 1100 I=1,KCRNT
         DO 1000 J=1,KCRNT
      CXRPQ(I,J)=CXPQN(I,J)+CRPQ1(I,J)
1000     CONTINUE
1100  CONTINUE
      DN=NA
      CALL DLSACG(ICRNT,CXRPQ,KCRNT,CIW,1,CKLN)
        IF ((IK .EQ. 1) .AND. (IR .EQ. 45)) THEN
      CALL KLCRNT(DN,CKLN,CIW)
        END IF
      CALL ESCFAR(NA,CKLN0,CKLN,CETHN,CEPHN)
            IF (RTHE .EQ. 0.D0)  THEN
      CETHN=COS(RPHI)*CETHN+SIN(RPHI)*CEPHN
      CEPHN=-SIN(RPHI)*CETHN+COS(RPHI)*CEPHN
            ELSEIF (RTHE .EQ. PI) THEN
      CETHN=-COS(RPHI)*CETHN-SIN(RPHI)*CEPHN
      CEPHN=-SIN(RPHI)*CETHN+COS(RPHI)*CEPHN
             END IF
C
      CESTH=CESTH+CETHN
      CESPH=CESPH+CEPHN
            IF (NA .NE. 0) THEN
                IF (NYSM .EQ. 0) THEN
      DO 1300 I=1,KCRNT
         DO 1200 J=1,KCRNT
      CXRPQ(I,J)=CXPQN(I,J)+CRPQ2(I,J)
1200     CONTINUE
1300  CONTINUE
                ELSE
      DO 1500 I=1,KCRNT
         DO 1400 J=1,KCRNT
      CXRPQ(I,J)=CXPQN(I,J)+CRPQ1(I,J)
1400     CONTINUE
1500  CONTINUE
               END IF
      CALL DLSACG(ICRNT,CXRPQ,KCRNT,CIW,1,CKLN)
               IF (IE .EQ. 1) THEN
```

```
      DO 1600 I=2,KCRNT,4
      I1=I+1
      CKLN(I)=-CKLN(I)
      CKLN(I1)=-CKLN(I1)
1600  CONTINUE
              END IF
              IF (IM .EQ. 1) THEN
      DO 1700 I=1,KCRNT,4
      I2=I+3
      CKLN(I)=-CKLN(I)
      CKLN(I2)=-CKLN(I2)
1700  CONTINUE
              END IF
      NA1=-NA
      DN=NA1
        IF ((IK .EQ. 1) .AND. (IS .EQ. 199)) THEN
      CALL KLCRNT(DN,CKLN,CIW)
        END IF
      CALL ESCFAR(NA1,CKLN0,CKLN,CETHN,CEPHN)
              IF (RTHE .EQ. 0.D0)  THEN
      CETHN=COS(RPHI)*CETHN+SIN(RPHI)*CEPHN
      CEPHN=-SIN(RPHI)*CETHN+COS(RPHI)*CEPHN
              ELSEIF (RTHE .EQ. PI) THEN
      CETHN=-COS(RPHI)*CETHN-SIN(RPHI)*CEPHN
      CEPHN=-SIN(RPHI)*CETHN+COS(RPHI)*CEPHN
               END IF
C
      CESTH=CESTH+CETHN
      CESPH=CESPH+CEPHN
            END IF
         END IF
1800  CONTINUE
C If the cylinder is coated with anisotropic material and we want to calculate
C equivalent currents K nad L on inner and outer surfaces.
         IF (IK .EQ. 1) THEN
            IF (IS .EQ. 199) THEN
      OPEN(31,FILE='khzz41.dz',STATUS='UNKNOWN')
      OPEN(32,FILE='lhzz41.dz',STATUS='UNKNOWN')
            END IF
      DO 1900 IF=-30,30
         DO 1850 JZ=-30,30
      AKPHPR=REAL(CKPHP(IF,JZ))
      AKPHPI=IMAG(CKPHP(IF,JZ))
      AKZPR=REAL(CKZP(IF,JZ))
      AKZPI=IMAG(CKZP(IF,JZ))
      AKPHNR=REAL(CKPHN(IF,JZ))
      AKPHNI=IMAG(CKPHN(IF,JZ))
      AKZNR=REAL(CKZN(IF,JZ))
      AKZNI=IMAG(CKZN(IF,JZ))
      ALPHPR=REAL(CLPHP(IF,JZ))
      ALPHPI=IMAG(CLPHP(IF,JZ))
      ALZPR=REAL(CLZP(IF,JZ))
      ALZPI=IMAG(CLZP(IF,JZ))
      ALPHNR=REAL(CLPHN(IF,JZ))
      ALPHNI=IMAG(CLPHN(IF,JZ))
      ALZNR=REAL(CLZN(IF,JZ))
      ALZNI=IMAG(CLZN(IF,JZ))
      WRITE(31,*) AKPHPR,'  ',AKPHPI
      WRITE(31,*) AKZPR,'  ',AKZPI
      WRITE(32,*) ALPHPR,'  ',ALPHPI
      WRITE(32,*) ALZPR,'  ',ALZPI
      WRITE(31,*) AKPHNR,'  ',AKPHNI
      WRITE(31,*) AKZNR,'  ',AKZNI
      WRITE(32,*) ALPHNR,'  ',ALPHNI
      WRITE(32,*) ALZNR,'  ',ALZNI
1850     CONTINUE
1900  CONTINUE
      CLOSE(31)
```

```
      CLOSE(32)
         END IF
      RETURN
C If incident angle is equal to zero, the program should go to this loop
C because it need to compute when n=+1 and n=-1 only.
2000  NA=1
      DN=NA
      CALL INCIDNT(NA,CIW0,CIW)
      CALL XPQN(NA)
C
         IF (IZ .EQ. 0) THEN
      CALL DLSACG(IXCRNT,CXPQN0,KXCRT,CIW0,1,CKLN0)
      CALL ESCFAR(NA,CKLN0,CKLN,CETHN,CEPHN)
C
C  If we calculate in X and Y components as theta approaches 0 or pi,
C  then theta component is cosin phi in X direction plus sine phi in Y
C  direction, and phi component is negative sine phi in X direction plus
C  cosin phi in Y direction.
         IF (RTHE .EQ. 0.D0)  THEN
      CETHN=COS(RPHI)*CETHN+SIN(RPHI)*CEPHN
      CEPHN=-SIN(RPHI)*CETHN+COS(RPHI)*CEPHN
         ELSEIF (RTHE .EQ. PI) THEN
      CETHN=-COS(RPHI)*CETHN-SIN(RPHI)*CEPHN
      CEPHN=-SIN(RPHI)*CETHN+COS(RPHI)*CEPHN
         END IF
C
      CESTH=CESTH+CETHN
      CESPH=CESPH+CEPHN
            IF (NA .NE. 0) THEN
               IF (IE .EQ. 1) THEN
      DO 2400 I=2,KXCRT,2
      CKLN0(I)=-CKLN0(I)
2400  CONTINUE
            END IF
            IF (IM .EQ. 1) THEN
      DO 2500 I=1,KXCRT,2
      CKLN0(I)=-CKLN0(I)
2500  CONTINUE
            END IF
      NA1=-NA
      DN=NA1
      CALL ESCFAR(NA1,CKLN0,CKLN,CETHN,CEPHN)
            IF (RTHE .EQ. 0.D0)  THEN
      CETHN=COS(RPHI)*CETHN+SIN(RPHI)*CEPHN
      CEPHN=-SIN(RPHI)*CETHN+COS(RPHI)*CEPHN
            ELSEIF (RTHE .EQ. PI) THEN
      CETHN=-COS(RPHI)*CETHN-SIN(RPHI)*CEPHN
      CEPHN=-SIN(RPHI)*CETHN+COS(RPHI)*CEPHN
             END IF
C
      CESTH=CESTH+CETHN
      CESPH=CESPH+CEPHN
            END IF
         END IF
C
         IF (IZ .EQ. 1) THEN
      DO 3100 I=1,KCRNT
      DO 3000 J=1,KCRNT
      CXRPQ(I,J)=CXPQN(I,J)+CRPQ1(I,J)
3000     CONTINUE
3100  CONTINUE
      DN=NA
      CALL DLSACG(ICRNT,CXRPQ,KCRNT,CIW,1,CKLN)
C
        IF ((IK .EQ. 1) .AND. (IR .EQ. 45)) THEN
      CALL KLCRNT(DN,CKLN,CIW)
        END IF
C
```

```
      CALL ESCFAR(NA,CKLN0,CKLN,CETHN,CEPHN)
           IF (RTHE .EQ. 0.D0)  THEN
      CETHN=COS(RPHI)*CETHN+SIN(RPHI)*CEPHN
      CEPHN=-SIN(RPHI)*CETHN+COS(RPHI)*CEPHN
           ELSEIF (RTHE .EQ. PI) THEN
      CETHN=-COS(RPHI)*CETHN-SIN(RPHI)*CEPHN
      CEPHN=-SIN(RPHI)*CETHN+COS(RPHI)*CEPHN
            END IF
C
      CESTH=CESTH+CETHN
      CESPH=CESPH+CEPHN
           IF (NA .NE. 0) THEN
                IF (NYSM .EQ. 0) THEN
      DO 3300 I=1,KCRNT
         DO 3200 J=1,KCRNT
      CXRPQ(I,J)=CXPQN(I,J)+CRPQ2(I,J)
3200     CONTINUE
3300  CONTINUE
                ELSE
      DO 3500 I=1,KCRNT
         DO 3400 J=1,KCRNT
      CXRPQ(I,J)=CXPQN(I,J)+CRPQ1(I,J)
3400     CONTINUE
3500  CONTINUE
                END IF
      CALL DLSACG(ICRNT,CXRPQ,KCRNT,CIW,1,CKLN)
                IF (IE .EQ. 1) THEN
      DO 3600 I=2,KCRNT,4
      I1=I+1
      CKLN(I)=-CKLN(I)
      CKLN(I1)=-CKLN(I1)
3600  CONTINUE
                END IF
                IF (IM .EQ. 1) THEN
      DO 3700 I=1,KCRNT,4
      I2=I+3
      CKLN(I)=-CKLN(I)
      CKLN(I2)=-CKLN(I2)
3700  CONTINUE
                END IF
      NA1=-NA
      DN=NA1
C
        IF ((IK .EQ. 1) .AND. (IS .EQ. 199)) THEN
      CALL KLCRNT(DN,CKLN,CIW)
        END IF
C
      CALL ESCFAR(NA1,CKLN0,CKLN,CETHN,CEPHN)
                IF (RTHE .EQ. 0.D0)  THEN
      CETHN=COS(RPHI)*CETHN+SIN(RPHI)*CEPHN
      CEPHN=-SIN(RPHI)*CETHN+COS(RPHI)*CEPHN
                ELSEIF (RTHE .EQ. PI) THEN
      CETHN=-COS(RPHI)*CETHN-SIN(RPHI)*CEPHN
      CEPHN=-SIN(RPHI)*CETHN+COS(RPHI)*CEPHN
                 END IF
C
      CESTH=CESTH+CETHN
      CESPH=CESPH+CEPHN
            END IF
         END IF
C
         IF (IK .EQ. 1) THEN
            IF (IS .EQ. 199) THEN
      OPEN(31,FILE='khzz41.dz',STATUS='UNKNOWN')
      OPEN(32,FILE='lhzz41.dz',STATUS='UNKNOWN')
            END IF
      DO 4200 IF=-30,30
         DO 4100 JZ=-30,30
```

```
      AKPHPR=REAL(CKPHP(IF,JZ))
      AKPHPI=IMAG(CKPHP(IF,JZ))
      AKZPR=REAL(CKZP(IF,JZ))
      AKZPI=IMAG(CKZP(IF,JZ))
      AKPHNR=REAL(CKPHN(IF,JZ))
      AKPHNI=IMAG(CKPHN(IF,JZ))
      AKZNR=REAL(CKZN(IF,JZ))
      AKZNI=IMAG(CKZN(IF,JZ))
      ALPHPR=REAL(CLPHP(IF,JZ))
      ALPHPI=IMAG(CLPHP(IF,JZ))
      ALZPR=REAL(CLZP(IF,JZ))
      ALZPI=IMAG(CLZP(IF,JZ))
      ALPHNR=REAL(CLPHN(IF,JZ))
      ALPHNI=IMAG(CLPHN(IF,JZ))
      ALZNR=REAL(CLZN(IF,JZ))
      ALZNI=IMAG(CLZN(IF,JZ))
      WRITE(31,*) AKPHPR,'  ',AKPHPI
      WRITE(31,*) AKZPR,'  ',AKZPI
      WRITE(32,*) ALPHPR,'  ',ALPHPI
      WRITE(32,*) ALZPR,'  ',ALZPI
      WRITE(31,*) AKPHNR,'  ',AKPHNI
      WRITE(31,*) AKZNR,'  ',AKZNI
      WRITE(32,*) ALPHNR,'  ',ALPHNI
      WRITE(32,*) ALZNR,'  ',ALZNI
4100       CONTINUE
4200   CONTINUE
      CLOSE(31)
      CLOSE(32)
         END IF
      RETURN
      END
CC
        SUBROUTINE INCIDNT(NA,CIW0,CIW)
C*********************************************************************
C This subroutine sets up the incident wave on an object which is coated
C with anisotropic material, or a perfect conductor.
C*********************************************************************
      INCLUDE 'REALTP.INC'
      INCLUDE 'CMPXTP.INC'
      INCLUDE 'MAINDM.INC'
C
      DIMENSION CIW0(KXCRT),CIW(KCRNT),DJNS(KNDIM1+1),DJPS(KQDIM1+2)
      COMMON /INPUT2/ IE,IM,THETAI,THESINI,THECOSI,RTHEI
      COMMON /INPUT4/ IZ,IK,IS,NYSM
      COMMON /INPUT3/ RTHE,RDELT,RPHI,RDELP,NTHTAO,NPHI,THESIN,THECOS,
     +                RHPI
      COMMON /RTHETA/ DL1COSI,DL2SINI,DLL
      COMMON /GCONST/ DKH,DKA,HH,HA,HSQ,DASQ,HSQN,DASQN,HHSQ,HDASQ,
     +         RAH,RAHSQ,DHA,DAH
      COMMON /CRNTDM/ NMAX,MXNG,IQMAX,IQMAX1,IQMAX2,IXCRNT,ICRNT,MXQEG,
     +                MXQOG
C
      NP=NA+1
      NP1=NP+1
      NM=NA-1
C
         IF (IZ .EQ. 0) GO TO 4000
C
C  This part computes the incident field on an anisotropic object
C
C  Initialize the column matrix of sum current on the anisotropic
      DO 100 IJ=1,KCRNT
      CIW(IJ)=CZERO
100    CONTINUE
C
C  If the incident angle is 90 degree or 0 degree
         IF (RTHEI .EQ. ZERO) GO TO 3000
      CALL DBSJNS(DL2SINI, KNDIM1, DJNS)
```

```
      DJN=DJNS(NP)
         IF (NA. EQ. 0) THEN
      DDJN=-DJNS(NP1)
         ELSE
      DDJN=HALF*(DJNS(NA)-DJNS(NP1))
         ENDIF
C Use IMSL library to compute Bessel's function series.
      CALL DBSJNS(DL1COSI,IQMAX2,DJPS)
      DO 600 IP=0,KQDIM
      NIP=IP+1
      DNIP=NIP
      NIP1=NIP+1
      NIP2=NIP+2
      N11=NA+IP-1
      N12=NA+IP
      IE1=MOD(N11,4)
      IE2=MOD(N12,4)
C
      CF1=CI2**IE1
      CF2=CI2**IE2
      IPW1=4*IP+1
      IPW2=IPW1+1
      IPW3=IPW2+1
      IPW4=IPW3+1
C
      DJP=DJPS(NIP)
      DJP1=DJPS(NIP1)
      DJP2=DJPS(NIP2)
         IF (IE .EQ. 1) THEN
      CEEPHI=CF1*(DJP+DJP2)*DDJN/DNIP
      CEHPHI=-TWO*NA*CF2*DJP1*DJN/DLL
      CEHZ=-CF2*THESINI*(DJP+DJP2)*DJN/DNIP
         ELSE
      CEEPHI=CZERO
      CEHPHI=CZERO
      CEHZ=CZERO
         END IF
C
         IF (IM .EQ. 1) THEN
      CMEPHI=TWO*NA*CF2*DJP1*DJN/DLL
      CMEZ=CF2*THESINI*(DJP+DJP2)*DJN/DNIP
      CMHPHI=CF1*(DJP+DJP2)*DDJN/DNIP
         ELSE
      CMEPHI=CZERO
      CMEZ=CZERO
      CMHPHI=CZERO
         END IF
C
      CIW(IPW1)=TWO*(CEEPHI+*CMEPHI)
      CIW(IPW2)=TWO*CMEZ
      CIW(IPW3)=TWO*(CEHPHI+CMHPHI)
      CIW(IPW4)=TWO*CEHZ
600   CONTINUE
      RETURN
C
3000  CALL DBSJNS(DKH,IQMAX2,DJPS)
      DO 3200 IP=0,KQDIM
      JP=IP+2
      DJP=DJPS(JP)
      IE11=MOD(IP,4)
      IE12=MOD(IP+1,4)
      CF1=CI2**IE11
      CF2=CI2**IE12
      IPW1=4*IP+1
      IPW3=IPW1+2
C
         IF (IE .EQ. 1) THEN
      CEEPHI=CF1*DJP/DKH
```

```
      CEHPHI=-CF2*DJP/DKH
         ELSE
      CEEPHI=CZERO
      CEHPHI=CZERO
         END IF
C
         IF (IM .EQ. 1) THEN
      CMEPHI=CF2*DJP/DKH
      CMHPHI=CF1*DJP/DKH
         ELSE
      CMEPHI=CZERO
      CMHPHI=CZERO
         END IF
C
      CIW(IPW1)=TWO*(CEEPHI+CMEPHI)
      CIW(IPW3)=TWO*(CEHPHI+CMHPHI)
3200  CONTINUE
      RETURN
C
C  This part compute incident fields on a perfect conductor
C  Initialize the column matrix of sum current on the conductor
4000  DO 4200 IX=1,KXCRT
      CIW0(IX)=CZERO
4200   CONTINUE
C
C  If the incident angle is 90 degree or 0 degree
         IF (RTHEI .EQ. ZERO) GO TO 6000
C
      CALL DBSJNS(DL2SINI, MXNG+1, DJNS)
      DJN=DJNS(NP)
         IF (NA .EQ. 0) THEN
      DDJN=-DJNS(NP1)
         ELSE
      DDJN=HALF*(DJNS(NA)-DJNS(NP1))
         ENDIF
C
      CALL DBSJNS(DL1COSI,IQMAX2,DJPS)
      DO 4400 IP=0,KQDIM
      NIP=IP+1
      DNIP=NIP
      NIP1=NIP+1
      NIP2=NIP+2
      N11=NA+IP-1
      N12=NA+IP
      IE1=MOD(N11,4)
      IE2=MOD(N12,4)
C
      CF1=CI2**IE1
      CF2=CI2**IE2
      IPW1=2*IP+1
      IPW2=IPW1+1
C
      DJP=DJPS(NIP)
      DJP1=DJPS(NIP1)
      DJP2=DJPS(NIP2)
         IF (IE .EQ. 1) THEN
      CEEPHI=CF1*(DJP+DJP2)*DDJN/DNIP
         ELSE
      CEEPHI=CZERO
         END IF
C
         IF (IM .EQ. 1) THEN
      CMEPHI=TWO*NA*CF2*DJP1*DJN/DLL
      CMEZ=CF2*THESINI*(DJP+DJP2)*DJN/DNIP
         ELSE
      CMEPHI=CZERO
      CMEZ=CZERO
         END IF
```

```
C
      CIW0(IPW1)=TWO*(CEEPHI+CMEPHI)
      CIW0(IPW2)=TWO*CMEZ
4400   CONTINUE
      RETURN
C
6000  CALL DBSJNS(DKH,IQMAX2,DJPS)
      DO 6200 IP=0,KQDIM
      JP=IP+2
      DJP=DJPS(JP)
      IE11=MOD(IP,4)
      IE12=MOD(IP+1,4)
      CF1=CI2**IE11
      CF2=CI2**IE12
      IPW1=2*IP+1
C
         IF (IE .EQ. 1) THEN
      CEEPHI=CF1*DJP/DKH
         ELSE
      CEEPHI=CZERO
         END IF
C
         IF (IM .EQ. 1) THEN
      CMEPHI=CF2*DJP/DKH
         ELSE
      CMEPHI=CZERO
         END IF
C
      CIW0(IPW1)=TWO*(CEEPHI+CMEPHI)
6200  CONTINUE
      RETURN
      END
C
      SUBROUTINE XPQ0
C*********************************************************************
C  This subroutine computes the matrix XN(P,Q) for N = 0 following a
C  call to XPQINI. This matrix is kept in the common block:
C     COMMON /XPQTMP/ CXPQN
C*********************************************************************
      INCLUDE 'REALTP.INC'
      INCLUDE 'CMPXTP.INC'
      INCLUDE 'MAINDM.INC'
C
      DIMENSION CXPQN(KCRNT,KCRNT),CXRPQ(KCRNT,KCRNT)
      DIMENSION CXPQN0(KXCRT,KXCRT)
      DIMENSION CGNE(0:MAXPEG,0:MAXPEG,KNDIM1+1),
     +           CGNO(0:MAXPOG,0:MAXPOG,KNDIM1+1)
      DIMENSION CDGNE(0:MAXPEG,0:MAXPEG,KNDIM1+1),
     +           CDGNO(0:MAXPOG,0:MAXPOG,KNDIM1+1)
      COMMON /GPQTMP/ CGNE,CDGNE,CGNO,CDGNO
      COMMON /XPQTMP/ CXPQN,CXRPQ,CXPQN0
      COMMON /CRNTDM/ NMAX,MXNG,IQMAX,IQMAX1,IQMAX2,IXCRNT,ICRNT,MXQEG,
     +                MXQOG
      COMMON /GCONST/ DKH,DKA,HH,HA,HSQ,DASQ,HSQN,DASQN,HHSQ,HDASQ,
     +               RAH,RAHSQ,DHA,DAH
      COMMON /INPUT4/ IZ,IK,IS,NYSM
      SAVE /XPQTMP/,/GPQTMP/,/GCONST/
C
      IF (N .NE. 0) THEN
      WRITE(*,*) 'Input N is not equal to 0 in XPQ0.'
      WRITE(*,*) 'Execution is stopped.'
      STOP
      END IF
C
         IF (IZ .EQ. 0) GO TO 4000
C
C  Initialize the XN(P,Q) matrix.
C  Null terms will be skipped later.
```

```
C
      DO 200 IQ=1,KCRNT
         DO 100 IP=1,KCRNT
      CXPQN(IP,IQ)=CZERO
100      CONTINUE
200   CONTINUE
C
C  Form the XN(P,Q) matrix for n = 0 and using on a perfect conductor
      CF31=DKA*CI1
C
      DO 1300 IQE=0,MXQEG-1
      IQE1=IQE+1
      IQ=2*IQE
      IQX1=4*IQ+1
      IQX2=IQX1+1
      IQX3=IQX2+1
      IQX4=IQX3+1
      DQ1=IQ+1
      FQ22=DQ1/HHSQ
         DO 1100 IPE=0,MXQEG-1
      IPE1=IPE+1
      IP=2*IPE
      IPX1=4*IP+1
      IPX2=IPX1+1
      IPX3=IPX2+1
      IPX4=IPX3+1
      DP1=IP+1
      F41=DKH/DP1
      CF11=F41*CF31
      F51=QUAR*F41*DKA
      CXPQN(IPX1,IQX1)=(CGNE(IPE1,IQE,2)-CGNE(IPE,IQE,2))*CF11
      CXPQN(IPX4,IQX1)=(CGNE(IPE1,IQE,2)-CGNE(IPE,IQE,2)+HA*(
     +                 CDGNE(IPE1,IQE,2)-CDGNE(IPE,IQE,2)))*F41
      CXPQN(IPX2,IQX2)=(TWO*FQ22*DP1*CGNO(IPE,IQE,1)+HALF*(
     +                 CGNE(IPE,IQE1,1)+CGNE(IPE1,IQE,1)-
     +                 CGNE(IPE1,IQE1,1)-CGNE(IPE,IQE,1)))*CF11
      CXPQN(IPX3,IQX2)=(CDGNE(IPE,IQE,1)+CDGNE(IPE1,IQE1,1)-
     +                 CDGNE(IPE1,IQE,1)-CDGNE(IPE,IQE1,1))*F51
      CXPQN(IPX2,IQX3)=-CXPQN(IPX4,IQX1)
      CXPQN(IPX3,IQX3)=CXPQN(IPX1,IQX1)
      CXPQN(IPX1,IQX4)=-CXPQN(IPX3,IQX2)
      CXPQN(IPX4,IQX4)=CXPQN(IPX2,IQX2)
1100     CONTINUE
1300  CONTINUE
      DO 2300 IQO=0,MXQOG-1
      IQO1=IQO+1
      IQ=2*IQO+1
      IQX1=4*IQ+1
      IQX2=IQX1+1
      IQX3=IQX2+1
      IQX4=IQX3+1
      DQ1=IQ+1
      FQ22=DQ1/HHSQ
         DO 2200 IPO=0,MXQOG-1
      IPO1=IPO+1
      IP=2*IPO+1
      IPX1=4*IP+1
      IPX2=IPX1+1
      IPX3=IPX2+1
      IPX4=IPX3+1
      DP1=IP+1
      F41=DKH/DP1
      CF11=F41*CF31
      F51=QUAR*F41*DKA
      CXPQN(IPX1,IQX1)=(CGNO(IPO1,IQO,2)-CGNO(IPO,IQO,2))*CF11
      CXPQN(IPX4,IQX1)=(CGNO(IPO1,IQO,2)-CGNO(IPO,IQO,2)+HA*(
     +                 CDGNO(IPO1,IQO,2)-CDGNO(IPO,IQO,2)))*F41
      CXPQN(IPX2,IQX2)=(TWO*FQ22*DP1*CGNE(IPO1,IQO1,1)+HALF*(
```

```
     +                   CGNO(IPO,IQO1,1)+CGNO(IPO1,IQO,1)-
     +                   CGNO(IPO1,IQO1,1)-CGNO(IPO,IQO,1)))*CF11
      CXPQN(IPX3,IQX2)=(CDGNO(IPO,IQO,1)+CDGNO(IPO1,IQO1,1)-
     +                   CDGNO(IPO1,IQO,1)-CDGNO(IPO,IQO1,1))*F51
      CXPQN(IPX2,IQX3)=-CXPQN(IPX4,IQX1)
      CXPQN(IPX3,IQX3)=CXPQN(IPX1,IQX1)
      CXPQN(IPX1,IQX4)=-CXPQN(IPX3,IQX2)
      CXPQN(IPX4,IQX4)=CXPQN(IPX2,IQX2)
2200     CONTINUE
2300  CONTINUE
      RETURN
C  Initialize the XN(P,Q) matrix.
C  Null terms will be skipped later.
C
4000  DO 4200 IQ=1,KXCRT
         DO 4100 IP=1,KXCRT
      CXPQN0(IP,IQ)=CZERO
4100     CONTINUE
4200   CONTINUE
C
C  Form the XN(P,Q) matrix for n = 0 and using on an anisotropic coat.
      CF31=DKA*CI1
C
      DO 4500 IQE=0,MXQEG-1
      IQE1=IQE+1
      IQ=2*IQE
      IQX1=2*IQ+1
      IQX2=IQX1+1
      DQ1=IQ+1
      FQ22=DQ1/HHSQ
         DO 4400 IPE=0,MXQEG-1
      IPE1=IPE+1
      IP=2*IPE
      IPX1=2*IP+1
      IPX2=IPX1+1
      DP1=IP+1
      F41=DKH/DP1
      CF11=F41*CF31
      F51=QUAR*F41*DKA
      CXPQN0(IPX1,IQX1)=(CGNE(IPE1,IQE,2)-CGNE(IPE,IQE,2))*CF11
      CXPQN0(IPX2,IQX2)=(TWO*FQ22*DP1*CGNO(IPE,IQE,1)+HALF*(
     +                   CGNE(IPE,IQE1,1)+CGNE(IPE1,IQE,1)-
     +                   CGNE(IPE1,IQE1,1)-CGNE(IPE,IQE,1)))*CF11
4400     CONTINUE
4500  CONTINUE
      DO 5300 IQO=0,MXQOG-1
      IQO1=IQO+1
      IQ=2*IQO+1
      IQX1=2*IQ+1
      IQX2=IQX1+1
      DQ1=IQ+1
      FQ22=DQ1/HHSQ
         DO 5200 IPO=0,MXQOG-1
      IPO1=IPO+1
      IP=2*IPO+1
      IPX1=2*IP+1
      IPX2=IPX1+1
      DP1=IP+1
      F41=DKH/DP1
      CF11=F41*CF31
      F51=QUAR*F41*DKA
      CXPQN0(IPX1,IQX1)=(CGNO(IPO1,IQO,2)-CGNO(IPO,IQO,2))*CF11
      CXPQN0(IPX2,IQX2)=(TWO*FQ22*DP1*CGNE(IPO1,IQO1,1)+HALF*(
     +                   CGNO(IPO,IQO1,1)+CGNO(IPO1,IQO,1)-
     +                   CGNO(IPO1,IQO1,1)-CGNO(IPO,IQO,1)))*CF11
5200     CONTINUE
5300  CONTINUE
      RETURN
```

```
      END
C
      SUBROUTINE XPQN(NIN)
C*********************************************************************
C  This subroutine calls the subroutine GDN to update G(P,Q,N) for
C  N = NIN+1, then forms the matrix XN(P,Q) for N = NIN > 0. This matrix
C  is kept in the common block:
C      COMMON /XPQTMP/ CXPQN
C*********************************************************************
C=====================================================================
C WARNING:
C IT IS ASSUMED THAT XPQINI AND XPQN FOR N FROM 1 TO NIN-1 HAVE BEEN
C CALLED SO THAT G(P,Q,N) AND ITS DERIVATIVE FOR N=NIN-1 AND N=NIN
C HAVE BEEN STORED IN THE COMMON BLOCK /GPQTMP/ WITH PROPER N-INDICES.
C=====================================================================
      INCLUDE 'REALTP.INC'
      INCLUDE 'CMPXTP.INC'
      INCLUDE 'MAINDM.INC'
C
      DIMENSION CXPQN(KCRNT,KCRNT),CXRPQ(KCRNT,KCRNT)
      DIMENSION CXPQN0(KXCRT,KXCRT)
      DIMENSION CGNE(0:MAXPEG,0:MAXPEG,KNDIM1+1),
     +                 CGNO(0:MAXPOG,0:MAXPOG,KNDIM1+1)
      DIMENSION CDGNE(0:MAXPEG,0:MAXPEG,KNDIM1+1),
     +                 CDGNO(0:MAXPOG,0:MAXPOG,KNDIM1+1)
      COMMON /GPQTMP/ CGNE,CDGNE,CGNO,CDGNO
      COMMON /XPQTMP/ CXPQN,CXRPQ,CXPQN0
      COMMON /CRNTDM/ NMAX,MXNG,IQMAX,IQMAX1,IQMAX2,IXCRNT,ICRNT,MXQEG,
     +                MXQOG
      COMMON /GCONST/ DKH,DKA,HH,HA,HSQ,DASQ,HSQN,DASQN,HHSQ,HDASQ,
     +               RAH,RAHSQ,DHA,DAH
      COMMON /NCONST/ DN,DNH,N
      COMMON /INPUT4/ IZ,IK,IS,NYSM
      SAVE /XPQTMP/,/GPQTMP/,/GCONST/
C
      N=NIN
      NA=ABS(N)
      IF (NA .LT. 1) THEN
      WRITE(*,*) 'Input ABS(N) is less than 1 in XPQN.'
      WRITE(*,*) 'Execution is stopped.'
      STOP
      END IF
C
      N0I=NA+1
      NPI=N0I+1
      NMI=NA
C
      DN0=NA
      DNP=N0I
      DNM=NA-1
C
         IF (IZ .EQ. 0) GO TO 4000
C
C  Initialize the XN(P,Q) matrix.
C  Null terms will be skipped later.
C
      DO 800 IQ=1,KCRNT
         DO 700 IP=1,KCRNT
      CXPQN(IP,IQ)=CZERO
700      CONTINUE
800   CONTINUE
C
C  Form the XN(P,Q) matrix.
      CF31=DKA*CI1
      F21=TWO*DN0
      FN11=F21*DN0/DASQ
C
      DO 1300 IQE=0,MXQEG-1
```

```
      IQE1=IQE+1
      IQ=2*IQE
      IQX1=4*IQ+1
      IQX2=IQX1+1
      IQX3=IQX2+1
      IQX4=IQX3+1
      DQ1=IQ+1
      FQ12=DN0*DQ1
      FQ22=DQ1/HHSQ
         DO 1100 IPE=0,MXQEG-1
      IPE1=IPE+1
      IP=2*IPE
      IPX1=4*IP+1
      IPX2=IPX1+1
      IPX3=IPX2+1
      IPX4=IPX3+1
      DP1=IP+1
      F41=HH/DP1
      CF11=F41*CF31
      F51=HALF*DKA*F41
      CXPQN(IPX1,IQX1)=((CGNE(IPE,IQE,N0I)-CGNE(IPE1,IQE,N0I))*FN11+
     +                  CGNE(IPE1,IQE,NMI)-CGNE(IPE,IQE,NMI)+
     +                  CGNE(IPE1,IQE,NPI)-CGNE(IPE,IQE,NPI))*CF11
      CXPQN(IPX4,IQX1)=((CGNE(IPE,IQE,NMI)-CGNE(IPE1,IQE,NMI))*DNM+
     +                 (CGNE(IPE1,IQE,NPI)-CGNE(IPE,IQE,NPI))*DNP+
     +                 HA*(CDGNE(IPE1,IQE,NMI)-CDGNE(IPE,IQE,NMI)+
     +                 CDGNE(IPE1,IQE,NPI)-CDGNE(IPE,IQE,NPI)))*F41
      CXPQN(IPX2,IQX2)=(FQ22*DP1*CGNO(IPE,IQE,N0I)+
     +                  CGNE(IPE,IQE1,N0I)+CGNE(IPE1,IQE,N0I)-
     +                  CGNE(IPE1,IQE1,N0I)-CGNE(IPE,IQE,N0I))*CF11
      CXPQN(IPX3,IQX2)=(CDGNE(IPE,IQE,N0I)+CDGNE(IPE1,IQE1,N0I)-
     +                  CDGNE(IPE1,IQE,N0I)-CDGNE(IPE,IQE1,N0I))*F51
      CXPQN(IPX2,IQX3)=-CXPQN(IPX4,IQX1)
      CXPQN(IPX3,IQX3)=CXPQN(IPX1,IQX1)
      CXPQN(IPX1,IQX4)=-CXPQN(IPX3,IQX2)
      CXPQN(IPX4,IQX4)=CXPQN(IPX2,IQX2)
1100     CONTINUE
         DO 1200 IPO=0,MXQOG-1
      IPO1=IPO+1
      IP=2*IPO+1
      IPX1=4*IP+1
      IPX2=IPX1+1
      IPX3=IPX2+1
      IPX4=IPX3+1
      DP1=IP+1
      F12=FQ12/DP1
      CXPQN(IPX2,IQX1)=F21*CGNE(IPO1,IQE,N0I)
      CXPQN(IPX3,IQX1)=(CGNE(IPO1,IQE,NMI)-CGNE(IPO1,IQE,NPI))*CF31
      CXPQN(IPX1,IQX2)=(CGNO(IPO1,IQE,N0I)-CGNO(IPO,IQE,N0I))*F12
      CXPQN(IPX1,IQX3)=-CXPQN(IPX3,IQX1)
      CXPQN(IPX4,IQX3)=CXPQN(IPX2,IQX1)
      CXPQN(IPX3,IQX4)=CXPQN(IPX1,IQX2)
1200     CONTINUE
1300  CONTINUE
      DO 2300 IQO=0,MXQOG-1
      IQO1=IQO+1
      IQ=2*IQO+1
      IQX1=4*IQ+1
      IQX2=IQX1+1
      IQX3=IQX2+1
      IQX4=IQX3+1
      DQ1=IQ+1
      FQ12=DN0*DQ1
      FQ22=DQ1/HHSQ
         DO 2100 IPE=0,MXQEG-1
      IPE1=IPE+1
      IP=2*IPE
      IPX1=4*IP+1
```

```
      IPX2=IPX1+1
      IPX3=IPX2+1
      IPX4=IPX3+1
      DP1=IP+1
      F12=FQ12/DP1
      CXPQN(IPX2,IQX1)=F21*CGNO(IPE,IQO,N0I)
      CXPQN(IPX3,IQX1)=(CGNO(IPE,IQO,NMI)-CGNO(IPE,IQO,NPI))*CF31
      CXPQN(IPX1,IQX2)=(CGNE(IPE1,IQO1,N0I)-CGNE(IPE,IQO1,N0I))*F12
      CXPQN(IPX1,IQX3)=-CXPQN(IPX3,IQX1)
      CXPQN(IPX4,IQX3)=CXPQN(IPX2,IQX1)
      CXPQN(IPX3,IQX4)=CXPQN(IPX1,IQX2)
2100     CONTINUE
         DO 2200 IPO=0,MXQOG-1
      IPO1=IPO+1
      IP=2*IPO+1
      IPX1=4*IP+1
      IPX2=IPX1+1
      IPX3=IPX2+1
      IPX4=IPX3+1
      DP1=IP+1
      F41=HH/DP1
      CF11=F41*CF31
      F51=HALF*DKA*F41
      CXPQN(IPX1,IQX1)=((CGNO(IPO,IQO,N0I)-CGNO(IPO1,IQO,N0I))*FN11+
     +                  CGNO(IPO1,IQO,NMI)-CGNO(IPO,IQO,NMI)+
     +                  CGNO(IPO1,IQO,NPI)-CGNO(IPO,IQO,NPI))*CF11
      CXPQN(IPX4,IQX1)=((CGNO(IPO,IQO,NMI)-CGNO(IPO1,IQO,NMI))*DNM+
     +                 (CGNO(IPO1,IQO,NPI)-CGNO(IPO,IQO,NPI))*DNP+
     +                 HA*(CDGNO(IPO1,IQO,NMI)-CDGNO(IPO,IQO,NMI)+
     +                 CDGNO(IPO1,IQO,NPI)-CDGNO(IPO,IQO,NPI)))*F41
      CXPQN(IPX2,IQX2)=(FQ22*DP1*CGNE(IPO1,IQO1,N0I)+
     +                  CGNO(IPO,IQO1,N0I)+CGNO(IPO1,IQO,N0I)-
     +                  CGNO(IPO1,IQO1,N0I)-CGNO(IPO,IQO,N0I))*CF11
      CXPQN(IPX3,IQX2)=(CDGNO(IPO,IQO,N0I)+CDGNO(IPO1,IQO1,N0I)-
     +                  CDGNO(IPO1,IQO,N0I)-CDGNO(IPO,IQO1,N0I))*F51
      CXPQN(IPX2,IQX3)=-CXPQN(IPX4,IQX1)
      CXPQN(IPX3,IQX3)=CXPQN(IPX1,IQX1)
      CXPQN(IPX1,IQX4)=-CXPQN(IPX3,IQX2)
      CXPQN(IPX4,IQX4)=CXPQN(IPX2,IQX2)
2200     CONTINUE
2300  CONTINUE
      RETURN
C  Initialize the XN(P,Q) matrix.
C  Null terms will be skipped later.
C
4000  DO 4800 IQ=1,KXCRT
         DO 4700 IP=1,KXCRT
      CXPQN0(IP,IQ)=CZERO
4700     CONTINUE
4800  CONTINUE
C
C  Form the XN(P,Q) matrix.
      CF31=DKA*CI1
      F21=TWO*DN0
      FN11=F21*DN0/DASQ
C
      DO 5300 IQE=0,MXQEG-1
      IQE1=IQE+1
      IQ=2*IQE
      IQX1=2*IQ+1
      IQX2=IQX1+1
      DQ1=IQ+1
      FQ12=DN0*DQ1
      FQ22=DQ1/HHSQ
         DO 5100 IPE=0,MXQEG-1
      IPE1=IPE+1
      IP=2*IPE
      IPX1=2*IP+1
```

```
      IPX2=IPX1+1
      DP1=IP+1
      F41=HH/DP1
      CF11=F41*CF31
      F51=HALF*DKA*F41
      CXPQN0(IPX1,IQX1)=((CGNE(IPE,IQE,N0I)-CGNE(IPE1,IQE,N0I))*FN11+
     +                   CGNE(IPE1,IQE,NMI)-CGNE(IPE,IQE,NMI)+
     +                   CGNE(IPE1,IQE,NPI)-CGNE(IPE,IQE,NPI))*CF11
      CXPQN0(IPX2,IQX2)=(FQ22*DP1*CGNO(IPE,IQE,N0I)+
     +                   CGNE(IPE,IQE1,N0I)+CGNE(IPE1,IQE,N0I)-
     +                   CGNE(IPE1,IQE1,N0I)-CGNE(IPE,IQE,N0I))*CF11
5100     CONTINUE
         DO 5200 IPO=0,MXQOG-1
      IPO1=IPO+1
      IP=2*IPO+1
      IPX1=2*IP+1
      IPX2=IPX1+1
      DP1=IP+1
      F12=FQ12/DP1
      CXPQN0(IPX2,IQX1)=F21*CGNE(IPO1,IQE,N0I)
      CXPQN0(IPX1,IQX2)=(CGNO(IPO1,IQE,N0I)-CGNO(IPO,IQE,N0I))*F12
5200     CONTINUE
5300  CONTINUE
      DO 6300 IQO=0,MXQOG-1
      IQO1=IQO+1
      IQ=2*IQO+1
      IQX1=2*IQ+1
      IQX2=IQX1+1
      DQ1=IQ+1
      FQ12=DN0*DQ1
      FQ22=DQ1/HHSQ
         DO 6100 IPE=0,MXQEG-1
      IPE1=IPE+1
      IP=2*IPE
      IPX1=2*IP+1
      IPX2=IPX1+1
      DP1=IP+1
      F12=FQ12/DP1
      CXPQN0(IPX2,IQX1)=F21*CGNO(IPE,IQO,N0I)
      CXPQN0(IPX1,IQX2)=(CGNE(IPE1,IQO1,N0I)-CGNE(IPE,IQO1,N0I))*F12
6100     CONTINUE
         DO 6200 IPO=0,MXQOG-1
      IPO1=IPO+1
      IP=2*IPO+1
      IPX1=2*IP+1
      IPX2=IPX1+1
      DP1=IP+1
      F41=HH/DP1
      CF11=F41*CF31
      F51=HALF*DKA*F41
      CXPQN0(IPX1,IQX1)=((CGNO(IPO,IQO,N0I)-CGNO(IPO1,IQO,N0I))*FN11+
     +                   CGNO(IPO1,IQO,NMI)-CGNO(IPO,IQO,NMI)+
     +                   CGNO(IPO1,IQO,NPI)-CGNO(IPO,IQO,NPI))*CF11
      CXPQN0(IPX2,IQX2)=(FQ22*DP1*CGNE(IPO1,IQO1,N0I)+
     +                   CGNO(IPO,IQO1,N0I)+CGNO(IPO1,IQO,N0I)-
     +                   CGNO(IPO1,IQO1,N0I)-CGNO(IPO,IQO,N0I))*CF11
6200     CONTINUE
6300  CONTINUE
      RETURN
      END
C
      SUBROUTINE ESCFAR(NIN,CKLN0,CKLN,CETHN,CEPHN)
C***************************************************************************
C  This subroutine computes the scattered fields in far zone
C
      INCLUDE 'REALTP.INC'
      INCLUDE 'CMPXTP.INC'
      INCLUDE 'MAINDM.INC'
```

```
C
      DIMENSION CKLN0(KXCRT)
      DIMENSION CKLN(KCRNT),DJNS(KNDIM1),DJPS(KQDIM1+2)
      COMMON /CRNTDM/ NMAX,MXNG,IQMAX,IQMAX1,IQMAX2,IXCRNT,ICRNT,MXQEG,
     +                MXQOG
      COMMON /GCONST/ DKH,DKA,HH,HA,HSQ,DASQ,HSQN,DASQN,HHSQ,HDASQ,
     +               RAH,RAHSQ,DHA,DAH
      COMMON /INPUT3/ RTHE,RDELT,RPHI,RDELP,NTHTAO,NPHI,THESIN,THECOS,
     +                RHPI
      COMMON /INPUT4/ IZ,IK,IS,NYSM
C
      N=NIN
      DN1=N
      NA=ABS(NIN)
      NP=NA+1
      NP1=NP+1
      PHI1=RPHI*N
      CI2N=CI2**N
      CENP=CONE*COS(PHI1)+CI1*SIN(PHI1)
      CETHN=CZERO
      CEPHN=CZERO
C
         IF ((RTHE .EQ. 0.) .OR. (RTHE .EQ. PI)) GO TO 2000
C
      TCOT=THECOS/THESIN
      DL2SIN=DKA*THESIN
      DL1COS=DKH*THECOS
      DNL2=DN1*TCOT/DKA
      CALL DBSJNS(DL2SIN, KNDIM1, DJNS)
C
      DJN=DJNS(NP)
         IF (N .EQ. 0) THEN
      DDJN=-DJNS(NP1)
         ELSE
      DDJN=HALF*(DJNS(NA)-DJNS(NP1))
         ENDIF
C
      KN=NA/2
      KNP=NP/2
         IF ((N .LT. 0) .AND. (KNP .GT. KN)) THEN
      DJN=-DJN
      DDJN=-DDJN
          ENDIF
C
      CALL DBSJNS(DL1COS, KQDIM1+2, DJPS)
C
         IF (IZ .EQ. 0) THEN
      DO 100 IP=0,IQMAX
      INP=IP+1
      NP2=INP+2
      N11=IP
      N12=IP+2
      IE1=MOD(N11,4)
      CF1=CI2**IE1
      IPW1=2*IP+1
      IPW2=IPW1+1
      DJP=DJPS(INP)
      DJP2=DJPS(NP2)
      DJPH=HALF*(DJP+DJP2)
      CETHN=CETHN+CI2N*DHA*CENP*CI1*CF1*DJN*(DNL2*CKLN0(IPW1)*DJP
     +       -THESIN*CKLN0(IPW2)*DJPH)
      CEPHN=CEPHN-CI2N*DHA*CENP*DDJN*CKLN0(IPW1)*CF1*DJP
 100  CONTINUE
         ELSE
      DO 200 IP=0,IQMAX
      INP=IP+1
      NP2=INP+2
      N11=IP
```

```
      N12=IP+2
      IE1=MOD(N11,4)
      CF1=CI2**IE1
      IPW1=4*IP+1
      IPW2=IPW1+1
      IPW3=IPW2+1
      IPW4=IPW3+1
      DJP=DJPS(INP)
      DJP2=DJPS(NP2)
      DJPH=HALF*(DJP+DJP2)
      CETHN=CETHN+CI2N*DHA*CENP*((CI1*DJN*DNL2*CKLN(IPW1)-DDJN
     +       *CKLN(IPW3))*DJP-CI1*DJN*THESIN*CKLN(IPW2)*DJPH)*CF1
      CEPHN=CEPHN-CI2N*DHA*CENP*((DDJN*CKLN(IPW1)+CI1*DJN*DNL2
     +       *CKLN(IPW3))*DJP-CI1*DJN*THESIN*CKLN(IPW4)*DJPH)*CF1
200   CONTINUE
         END IF
      RETURN
C
2000      IF (NA .NE. 1) THEN
      GO TO 3000
          END IF
      CALL DBSJNS(DKH,KQDIM1,DJPS)
C
         IF (IZ .EQ. 0) THEN
      DO 2100 IP=0,IQMAX
      INP=IP+1
      IPW1=2*IP+1
      JP=MOD(IP,4)
      CF1=CI1**JP
      CF2=CI2**JP
C
      DJP=DJPS(INP)
            IF (RTHE .EQ. 0.) THEN
               IF (N .LT. 0) THEN
      CETHN=CETHN-DAH*DJP*CF2*CKLN0(IPW1)
      CEPHN=CEPHN+DAH*DJP*CF2*CI1*CKLN0(IPW1)
               ELSE
      CETHN=CETHN+DAH*DJP*CF2*CKLN0(IPW1)
      CEPHN=CEPHN+DAH*DJP*CF2*CI1*CKLN0(IPW1)
               END IF
            END IF
            IF (RTHE .EQ. PI) THEN
               IF (N .LT. 0) THEN
      CETHN=CETHN-DAH*DJP*CF1*CKLN0(IPW1)
      CEPHN=CEPHN+DAH*DJP*CF1*CI1*CKLN0(IPW1)
               ELSE
      CETHN=CETHN+DAH*DJP*CF1*CKLN0(IPW1)
      CEPHN=CEPHN+DAH*DJP*CF1*CI1*CKLN0(IPW1)
               END IF
            END IF
2100  CONTINUE
         ELSE
      DO 2200 IP=0,IQMAX
      INP=IP+1
      IPW1=4*IP+1
      IPW3=IPW1+2
      JP=MOD(IP,4)
      CF1=CI1**JP
      CF2=CI2**JP
C
      DJP=DJPS(INP)
            IF (RTHE .EQ. 0.) THEN
               IF (N .LT. 0) THEN
      CETHN=CETHN-DAH*DJP*CF2*(CKLN(IPW1)-CI1*CKLN(IPW3))
      CEPHN=CEPHN+DAH*DJP*CF2*(CI1*CKLN(IPW1)+CKLN(IPW3))
               ELSE
      CETHN=CETHN+DAH*DJP*CF2*(CKLN(IPW1)+CI1*CKLN(IPW3))
      CEPHN=CEPHN+DAH*DJP*CF2*(CI1*CKLN(IPW1)-CKLN(IPW3))
```

```
             END IF
          END IF
          IF (RTHE .EQ. PI) THEN
             IF (N .LT. 0) THEN
      CETHN=CETHN-DAH*DJP*CF1*(CKLN(IPW1)+CI1*CKLN(IPW3))
      CEPHN=CEPHN+DAH*DJP*CF1*(CI1*CKLN(IPW1)-CKLN(IPW3))
             ELSE
      CETHN=CETHN+DAH*DJP*CF1*(CKLN(IPW1)-CI1*CKLN(IPW3))
      CEPHN=CEPHN+DAH*DJP*CF1*(CI1*CKLN(IPW1)+CKLN(IPW3))
             END IF
          END IF
2200  CONTINUE
         END IF
3000  RETURN
      END
C
        SUBROUTINE KLCRNT(DN,CKLN,CIW)
C***********************************************************************
C  This subroutine computes equivalent currents K and L on inner and outer
C  surfaces respectively.
      INCLUDE 'REALTP.INC'
      INCLUDE 'CMPXTP.INC'
      INCLUDE 'MAINDM.INC'
C
      DIMENSION CKLN(KCRNT),CIW(KCRNT)
      DIMENSION CKLNP(KCRNT),CKLNN(KCRNT)
      DIMENSION CRPQ31(KCRNT,KCRNT),CRPQ32(KCRNT,KCRNT)
      DIMENSION CKPHP(61,61),CKZP(61,61),CLPHP(61,61),
     +          CLZP(61,61), CKPHN(61,61),CKZN(61,61),
     +          CLPHN(61,61),CLZN(61,61)
      COMMON /CKLMTX/ CKPHP,CKZP,CLPHP,CLZP,CKPHN,CKZN,CLPHN,CLZN
      COMMON /XPQTMP2/ CRPQ31,CRPQ32
      COMMON /CRNTDM/ NMAX,MXNG,IQMAX,IQMAX1,IQMAX2,IXCRNT,ICRNT,MXQEG,
     +                MXQOG
      COMMON /INPUT3/ RTHE,RDELT,RPHI,RDELP,NTHTAO,NPHI,THESIN,THECOS,
     +                RHPI
      COMMON /INPUT4/ IZ,IK,IS,NYSM
C
      DO 400 IX=1,ICRNT
      CKLNP(IX)=CZERO
      CKLNN(IX)=CZERO
400    CONTINUE
      PHI1=DN*RPHI
      CPHI=CONE*COS(PHI1)+CI1*SIN(PHI1)
C
      DO 600 IX=1,ICRNT
         DO 500 IY=1,ICRNT
      CKLNP(IX)=CKLNP(IX)+CRPQ31(IX,IY)*CKLN(IY)
500      CONTINUE
600   CONTINUE
C
      DO 800 IX=1,ICRNT
         DO 700 IY=1,ICRNT
      CKLNN(IX)=CKLNN(IX)+CRPQ32(IX,IY)*CKLN(IY)
700      CONTINUE
800   CONTINUE
C
      DO 1600 IF=-30,30
      DF=IF
      DPHI=DF*PI/32.D0
      DNPHI=DN*DPHI
      CPHI=CONE*DCOS(DNPHI)+CI1*DSIN(DNPHI)
         DO 1500 JZ=-30,30
      DJZ=JZ
      DZ=DJZ/32.D0
      DV=DACOS(DZ)
      SINV=DSIN(DV)
```

```
C
      CKPHP(IF,JZ)=CZERO
      CKZP(IF,JZ)=CZERO
      CLPHP(IF,JZ)=CZERO
      CLZP(IF,JZ)=CZERO
      CKPHN(IF,JZ)=CZERO
      CKZN(IF,JZ)=CZERO
      CLPHN(IF,JZ)=CZERO
      CLZN(IF,JZ)=CZERO
C
            DO 1400 IP=0,IQMAX
      P=IP
      P1=IP+1
      DPV=P*DV
      DP1V=P1*DV
      DCSPV=DCOS(DPV)
      SINP1V=DSIN(DP1V)
      IPX1=IP*4+1
      IPX2=IPX1+1
      IPX3=IPX2+1
      IPX4=IPX3+1
      CKPHP(IF,JZ)=CKPHP(IF,JZ)+CPHI*(CKLNP(IPX1)+CIW(IPX4))
     +             *DCSPV/PI/SINV
      CKZP(IF,JZ)=CKZP(IF,JZ)+CPHI*(CKLNP(IPX2)-CIW(IPX3))*SINP1V/PI
      CLPHP(IF,JZ)=CLPHP(IF,JZ)+CPHI*(CKLNP(IPX3)-CIW(IPX2))
     +             *DCSPV/PI/SINV
      CLZP(IF,JZ)=CLZP(IF,JZ)+CPHI*(CKLNP(IPX4)+CIW(IPX1))*SINP1V/PI
      CKPHN(IF,JZ)=CKPHN(IF,JZ)+CPHI*(CKLNN(IPX1)-CIW(IPX4))
     +             *DCSPV/PI/SINV
      CKZN(IF,JZ)=CKZN(IF,JZ)+CPHI*(CKLNN(IPX2)+CIW(IPX3))*SINP1V/PI
      CLPHN(IF,JZ)=CLPHN(IF,JZ)+CPHI*(CKLNN(IPX3)+CIW(IPX2))
     +             *DCSPV/PI/SINV
      CLZN(IF,JZ)=CLZN(IF,JZ)+CPHI*(CKLNN(IPX4)-CIW(IPX1))*SINP1V/PI
1400        CONTINUE
1500     CONTINUE
1600  CONTINUE
      RETURN
      END
```

J. SUBROUTINE RCSPAREA

```
      SUBROUTINE RCSPAREA(CESTH,CESPH)
C*************************************************************************
C  This subroutine computes cross section per projected area in all direction.
C
      INCLUDE 'REALTP.INC'
      INCLUDE 'CMPXTP.INC'
C
      REAL*4 ARCSPPA1,ARCSPPA2,APHASE1,APHASE2
      COMMON /GCONST/ DKH,DKA,HH,HA,HSQ,DASQ,HSQN,DASQN,HHSQ,HDASQ,
     +         RAH,RAHSQ,DHA,DAH
      COMMON /INPUT3/ RTHE,RDELT,RPHI,RDELP,NTHTAO,NPHI,THESIN,THECOS,
     +                RHPI
C
      ESTH=ABS(CESTH)
      ESPH=ABS(CESPH)
      ESTHSQ=ESTH*ESTH
      ESPHSQ=ESPH*ESPH
         IF (RTHE .EQ. RHPI) THEN
      RCSPPA1=PI*ESTHSQ/DKH/DKA
      RCSPPA2=PI*ESPHSQ/DKH/DKA
         ELSE IF ((RTHE .EQ. ZERO) .OR. (RTHE .EQ. PI)) THEN
      RCSPPA1=4.D0*ESTHSQ/DASQ
      RCSPPA2=4.D0*ESPHSQ/DASQ
         ELSE
```

```
      AP=ABS(QUAR*DASQ*THECOS)+ABS(DKH*DKA*THESIN/PI)
      RCSPPA1=ESTHSQ/AP
      RCSPPA2=ESPHSQ/AP
         END IF
      ARCSPPA1=RCSPPA1
      ARCSPPA2=RCSPPA2
C
      ER1=REAL(CESTH)
      EI1=IMAG(CESTH)
      ER2=REAL(CESPH)
      EI2=IMAG(CESPH)
      PHASE1=DATAN2(EI1,ER1)
      PHASE2=DATAN2(EI2,ER2)
      APHASE1=PHASE1
      APHASE2=PHASE2
      WRITE (21,*) ARCSPPA1,APHASE1,ARCSPPA2,APHASE2
C
      RETURN
      END
```

LIST OF REFERENCES

[1] M. E. Morris, private communication.

[2] J. A. Stratton and L. J. Chu, "Diffraction Theory of Electromagnetic Waves." *Physical Rev.,* Vol. 56, pp.99-107, 1939.

[3] C. Müller, *Foundations of the Mathematical Theory of Electromagnetic Waves*, Springer-Verlag, 1969.

[4] V. H. Weston, "Theory of Absorbers in Scattering,"*IEEE Trans. Antennas Propagat.*, Vol 11, 1963.

[5] R. E. Collin, *Field Theory of Guided Waves,* Sec. 1.5, IEEE Press, 1991.

[6] H.-M. Lee, "Electromagnetic Scattering of Tubular Cylindrical Structure--Double Series Formulation and Some Results,"*IEEE Trans. Antennas Propagat.,* Vol. AP-35, No.4, pp. 384-390 1987.

[7] H.-M. Lee, C. P. Chung, D. Geller and B.Haklay, "H_{11} Circular Waveguide Mode and Back Scattering Cross Section Along the Axis of a Thin Walled Tubular Cylinder of Finite Length," *IEE Proceedings,* Vol. 133, Pt. H., pp. 77-78, 1986

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center 8725 John J. Kingman Rd., STE 0944 Ft. Belvoir, VA 22060-6218	2
2.	Dudlley Knox Library, Naval Postgraduate School 411 Dyer Rd. Monterey, CA 93943-5101	2
3.	Chairman, Code EC Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5121	1
4.	Hung-Mou Lee, EC/Lh Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5121	2
5.	David C. Jenn, EC/Jn Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5121	1
6.	Sherif Michael,EC/Mi Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5121	1
7.	Peter C. Chu, OC/Cu Department of Cceanography Naval Postgraduate School Monterey, CA 93943-5121	1
8.	Fariba Fahroo, MA/Ff Department of Mathematics Naval Postgraduate School	1

Monterey, CA 93943-5121

9. Dr. Marvine E. Morris — 1
Sandia National Laboratories
Organization 2753
P. O. Box 5800
Albuquerque, NM 87158

10. Chen-Kuo Yu — 2
4F-2, 282, Tong-Men Rd
Ku-Shan Distr., Kaohsung
Taiwan, ROC

11. Library — 1
Naval Academy
Tsoyn, Kaohsung
Taiwan, ROC